16

KARLSRUHER BEITRÄGE ZUR REGELUNGS- UND STEUERUNGSTECHNIK

Florian Köpf

Adaptive Dynamic Programming: Solltrajektorienfolgeregelung und Konvergenzbedingungen



Florian Köpf

Adaptive Dynamic Programming:
Solltrajektorienfolgeregelung und Konvergenzbedingungen

Karlsruher Beiträge zur
Regelungs- und Steuerungstechnik
Karlsruher Institut für Technologie

Band 16

Adaptive Dynamic Programming: Solltrajektorienfolgeregelung und Konvergenzbedingungen

von
Florian Köpf



Karlsruher Institut für Technologie
Institut für Regelungs- und Steuerungssysteme

Adaptive Dynamic Programming:
Solltrajektorienfolgeregelung und Konvergenzbedingungen

Zur Erlangung des akademischen Grades eines Doktor-Ingenieurs
von der KIT-Fakultät für Elektrotechnik und Informationstechnik des
Karlsruher Instituts für Technologie (KIT) genehmigte Dissertation

von Florian Köpf, M.Sc.

Tag der mündlichen Prüfung: 13. Januar 2022
Hauptreferent: Prof. Dr.-Ing. Sören Hohmann
Korreferent: Prof. Dr.-Ing. Daniel Görges

Impressum



Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark
of Karlsruhe Institute of Technology.
Reprint using the book cover is not allowed.

www.ksp.kit.edu

Print on Demand 2022 – Gedruckt auf FSC-zertifiziertem Papier

ISSN 2511-6312
ISBN 978-3-7315-1193-9
DOI 10.5445/KSP/1000145970

Danksagung

Diese Dissertation entstand während meiner Zeit als wissenschaftlicher Mitarbeiter am Institut für Regelungs- und Steuerungssysteme (IRS) am Karlsruher Institut für Technologie (KIT). Für die dort gebotene Gelegenheit zur fachlichen und persönlichen Weiterentwicklung möchte ich mich an dieser Stelle herzlich bedanken.

Besonderer Dank gilt meinem Hauptreferenten Prof. Dr.-Ing. Sören Hohmann für seine ausgezeichnete Betreuung. Die zahlreichen anregenden Diskussionen und angenehmen Gespräche haben mir stets neue Denkanstöße geliefert und das mir entgegengebrachte Vertrauen und die gebotenen Freiheiten haben wesentlich zum Gelingen dieser Arbeit beigetragen. Herzlich danke ich auch Prof. Dr.-Ing. Daniel Görges für die freundliche Übernahme des Korreferats, spannende Gespräche und das Interesse an der vorliegenden Arbeit.

Der gemeinsame Austausch mit meinen Kolleginnen und Kollegen am IRS und die gegenseitige Unterstützung haben für eine bemerkenswerte Arbeitsatomsphäre gesorgt. Neben unzähligen fachlichen Diskussionen haben auch die persönlichen Gespräche und gemeinsamen Erlebnisse ein inspirierendes Umfeld erschaffen. Besonderer Dank hierfür gilt Jairo Inga und Michael Flad, die mich als Gruppenleiter über fachliche Belange hinaus bei meiner Tätigkeit als wissenschaftlicher Mitarbeiter unterstützt haben. Esther Bischoff, Christian Braun, Philipp Karg, Simon Rothfuß, Julian Schneider und abermals Jairo Inga, mit denen teilweise gemeinsame Publikationen entstanden sind, haben wertvolle Anmerkungen zur vorliegenden Arbeit geliefert. Lukas Kölsch, Simon Rothfuß und Manuel Schwartz haben daneben den Arbeitsalltag in der „Legebatterie" bereichert, während Mathias Kluwe und Beate Stassen durch ihre Hilfsbereitschaft und herzliche Art wichtige Säulen des Instituts darstellen. Dankbar bin ich auch Luca Puccetti, der mich im Rahmen eines gemeinsamen Forschungsprojekts in Aschheim auf die Teststrecke mitgenommen hat und der immer für fachliche Diskussionen zu begeistern war, sowie Sean Kille für seine Unterstützung am Ball-auf-Platte-System.

Ganz herzlich bedanke ich mich zudem bei allen Studierenden, deren Abschlussarbeiten ich im Rahmen meiner Tätigkeit am IRS betreuen durfte. Die Zusammenarbeit hat mir dabei viel Freude bereitet und der hierbei geführte fachliche Austausch lieferte wertvolle Beiträge zum Erfolg der vorliegenden Dissertation.

Für den gebotenen Ausgleich während der letzten Jahre danke ich meinen Freunden und meiner Familie von Herzen. Meine Eltern Karin und Jürgen sind immer für mich da und unterstützen mich jederzeit uneingeschränkt. Katharina hat mich unermüdlich zwischen all den Formeln und Beweisen geerdet und ihr gewissenhaftes Lektorat der vorliegenden Arbeit hat den letzten Feinschliff gegeben. Danke, dass du für mich da bist.

Karlsruhe, im Januar 2022

The outcome of any serious research can only be to make two questions grow where only one grew before.

Thorstein Bunde Veblen

Kurzfassung

Adaptive Dynamic Programming (ADP) steht als vielversprechendes und zukunftsorientiertes regelungstechnisches Werkzeug im Fokus der aktuellen Forschung. Allerdings existieren hierfür bislang weder flexibel einsetzbare, mit dem ADP-Mechanismus kompatible Solltrajektoriendarstellungen noch theoretische Untersuchungen hinsichtlich einer geeigneten Systemanregung zur Sicherstellung der Konvergenz.

Die vorliegende Arbeit schließt diese Lücken: Zum einen werden erstmals zeitdiskrete und zeitkontinuierliche Methoden präsentiert und analysiert, die flexible Solltrajektoriendarstellungen in ADP-Ansätze integrieren. Die explizite Abhängigkeit der vorgestellten, neuartigen Value- bzw. Q-Function und des darauf basierenden gelernten Regelgesetzes von Trajektorienparametern, die den aktuellen Sollverlauf repräsentieren, ermöglicht eine variable Vorgabe der Solltrajektorie zur Laufzeit. Zum anderen werden erstmalig theoretische Bedingungen an den Systemzustand hergeleitet, die sicherstellen, dass eine für die Konvergenz der Adaption zentrale Anregungseigenschaft erfüllt ist. Verbleibende Freiheitsgrade erlauben zudem die Berücksichtigung anwendungsspezifischer Anforderungen bei der Systemanregung. Die theoretischen Aussagen werden in Simulationen bestätigt.

Erste reale Anwendungen der vorgestellten adaptiven optimalen Trajektorienfolgeregelungsmethoden offenbaren schließlich das Potenzial dieser Ansätze. Flexible und effiziente Regler, die aufgrund der Berücksichtigung des Solltrajektorienverlaufs vorausschauend agieren, können ohne aufwendige Modellbildung aus realen Messdaten erlernt werden und sind zudem bisherigen Ansätzen bezüglich ihrer Performanz überlegen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungen und Symbole

Abkürzungen

Abkürzung	Beschreibung
ADP	Adaptive Dynamic Programming
BINF	Byrnes-Isidori-Normalform
CAN	Controller Area Network
DPG	Deterministic Policy Gradient
FIR	Finite Impulse Response
GUI	Graphical User Interface
HDP	Heuristic Dynamic Programming
HJB	Hamilton-Jacobi-Bellman
IRL	Integral Reinforcement Learning
LQ	linear-quadratisch
LSPI	Least-Squares Policy Iteration
o. B. d. A.	ohne Beschränkung der Allgemeinheit
PE	persistently excited
PI	Policy Iteration
PRADP	Parametrized Reference Adaptive Dynamic Programming
QEP	Quadrature Encoder Pulse
RL	Reinforcement Learning
SR	sufficiently rich
TD	temporal difference
u. d. N.	unter der Nebenbedingung
USB	Universal Serial Bus
VI	Value Iteration

Lateinische Buchstaben

Symbol	Beschreibung
$\boldsymbol{A}$	autonomer Systemanteil bei linearem System
a	Amplituden von Sinusfunktionen bzw. Beschleunigung
$\boldsymbol{B}$	Eingangsdynamikmatrix bei linearem System

Symbol	Beschreibung
b	Frequenzvorfaktor
$\boldsymbol{C}_z$	Matrizen zur Definition der Frequenzbedingungen Ω
c	Amplituden Kosinusfunktionen
c_{norm}	Normierungsfaktor
$\boldsymbol{D}$	Dynamikmatrix Referenzparameter bzw. Entkopplungsmatrix
d	Frequenzvorfaktor bzw. Polynomgrad
$\boldsymbol{e}$	Einheitsvektor bzw. Abweichung vom Sollzustand
e	Restterm
$e_{\hat{\boldsymbol{w}}}$	Schwellwert für Abbruchbedingung des LSPI-Algorithmus
$\boldsymbol{F}(\cdot)$	allgemeine Systemdynamik
$\boldsymbol{F}_{\text{ref}}$	Dynamikmatrix des Sollzustands $\boldsymbol{x}_\text{r}$
$\boldsymbol{f}(\cdot)$	autonomer Systemanteil
$\boldsymbol{f}_{\boldsymbol{x}_\text{r}}(\cdot)$	Dynamik des Sollzustands $\boldsymbol{x}_\text{r}$
$\boldsymbol{f}_{\boldsymbol{x}_\text{r},\boldsymbol{x}}(\cdot)$	Abbildung zwischen Systemzustand $\boldsymbol{x}$ und Sollzustand $\boldsymbol{x}_\text{r}$
$\boldsymbol{f}_{\boldsymbol{\zeta}}(\cdot)$	Dynamik des Referenzparameters $\boldsymbol{\zeta}$
$\boldsymbol{f}_{\boldsymbol{x}_\text{r},\boldsymbol{\zeta}}(\cdot)$	Abbildung zwischen Referenzparameter $\boldsymbol{\zeta}$ und Sollzustand $\boldsymbol{x}_\text{r}$
f	Potenzen der Basisfunktionsmonome
$\bar{f}$	Amplituden der Basisfunktionselemente
$\boldsymbol{g}(\cdot)$	Eingangsfunktion der Systemdynamik
g	Amplituden Sinusfunktionen
$\bar{g}$	Amplituden Kosinusfunktionen
$H(\cdot)$	Hamilton-Funktion
h	Dimension des Gewichtsvektors $\boldsymbol{w}$
$\bar{h}$	Laufindex
h_{FIR}	Länge des FIR-Filters
$\boldsymbol{I}, \boldsymbol{I}_n$	Einheitsmatrix bzw. Einheitsmatrix der Dimension $n \times n$
$I^{[X]}, I^{[Y]}$	Motorstrom in X- bzw. Y-Richtung
i	Laufindex
$J(\cdot)$	Gütefunktional
j	Laufindex
j	imaginäre Einheit
$\boldsymbol{K}$	Reglermatrix
K	Summenobergrenze
k	diskreter Zeitschritt bzw. Laufindex
L	Summenobergrenze
l	Iterationsindex/Trainingsschritt bzw. Laufindex
$\boldsymbol{M}$	Matrix
M	Sampleanzahl Training
M_B	Batchgröße Training
m	Anzahl an Frequenzen bzw. Index
N	Anzahl der Spieler/Regler
N_P	Anzahl verschiedener Frequenztupel $\mathcal{P}_i$

Symbol	Beschreibung
n	Systemordnung
n_{h}	Vorausschauhorizontlänge der Solltrajektorie
n_{z}	Dimension von $\boldsymbol{z}$
n_{ζ}	Dimension des Referenzparameters $\boldsymbol{\zeta}$
o	Laufindex
$\boldsymbol{P}$	Riccati-Matrix
$\boldsymbol{P}^{\mathrm{f}}$	Frequenzkombinationen
$\boldsymbol{P}_{\mathrm{LS}}$	Projektionsmatrix der (gewichteten) Least-Squares-Regression
p	Anzahl der Eingangsgrößen
$\boldsymbol{Q}$	Matrix zur quadratischen Bestrafung des Systemzustands
$Q(\cdot)$	Q-Funktion
Q_{y}	Faktor zur quadratischen Bestrafung des Ausgangsfehlers
$q(\cdot)$	Bestrafungsfunktion Systemzustand
$\boldsymbol{R}$	Matrix zur quadratischen Bestrafung der Stellgröße
$\boldsymbol{r}$	Einschrittkostenvektor
$r(\cdot)$	Einschrittkosten
$\boldsymbol{S}$	Projektionsmatrix
$\boldsymbol{s}$	Projektionsvektor
$s^{[X]}, s^{[Y]}$	Ballposition in X- bzw. Y-Richtung
s_{r}	Soll-Ballposition in ADP-kompatibler Form
$s_{\mathrm{r,soll}}$	Soll-Ballposition
$\boldsymbol{T}^{\mathrm{f}}$	trigonometrischer Typ zu Frequenzkombinationen $\boldsymbol{P}^{\mathrm{f}}$
T	Zeitpunkt
T_{d}	Konstante für zeitdiskrete PE-Definition
T_{IRL}	Integrationsdauer des Integral-Reinforcement-Learning-Intervalls
T_{konv}	Konvergenzzeit der Policy Iteration
T_{sim}	Simulationsdauer
T_{t}	Schwellwert
t	Zeit
Δt	Abtastzeit
$\boldsymbol{u}$	Stellgröße
$\boldsymbol{u}_{\mathrm{ex}}$	Anregungssignal
$u_{\mathrm{ex},1,\mathrm{w}}$	Anregungssignal weißes Rauschen
$V(\cdot)$	Value Function
$\boldsymbol{v}_{\mathrm{freq}}$	Vektor aus Sinus- und Kosinusfunktionen
$v^{[X]}, v^{[Y]}$	Ballgeschwindigkeit in X- bzw. Y-Richtung
$\boldsymbol{W}_{\mathrm{p}}$	Gewichtungsmatrix der gewichteten Least-Squares-Regression
$W_{\bar{h}}$	Frequenzmenge
$\boldsymbol{w}$	Gewichtsvektor
$\boldsymbol{w}_{\mathrm{FIR}}$	Gewichtsvektor FIR-Filter
$\boldsymbol{X}$	Hilfsmatrix
X	Plattendimension in X-Richtung

Symbol	Beschreibung
$\boldsymbol{x}$	Systemzustand
$\boldsymbol{x}_0$	Anfangszustand
$\boldsymbol{x}_{\mathrm{r}}$	Sollzustand in ADP-kompatibler Form
$\boldsymbol{x}_{\mathrm{r,soll}}$	Sollzustand
$\boldsymbol{x}_{\mathrm{PE}}$	Anregungstrajektorie
$\tilde{\boldsymbol{x}}$	erweiterter Zustand
$\tilde{\boldsymbol{x}}^{\diamond}$	erweiterter Zustand mit Anregungsrauschen auf Referenzparameter
$\tilde{\boldsymbol{x}}^{\diamond\diamond}$	erweiterter Zustand mit kompatiblem Referenzparameter
$\bar{x}$	Hilfszustand
$\boldsymbol{Y}$	Hilfsmatrix
Y	Plattendimension in Y-Richtung
$\boldsymbol{y}$	realer oder fiktiver Systemausgang
$\bar{\boldsymbol{y}}, \bar{\bar{\boldsymbol{y}}}, \tilde{\boldsymbol{y}}$	Hilfsvektoren
y	Ausgang bzw. laterale Position
y_{r}	Sollausgang in ADP-kompatibler Form
$y_{\mathrm{r,soll}}$	Sollausgang
$\boldsymbol{Z}$	Solltrajektorienparametermatrix bzw. Hilfsmatrix
$\boldsymbol{z}_{\mathrm{B}}$	Systemzustand in BINF
$\boldsymbol{z}_{k,i}$	Solltrajektorienparameter für Zustand i im Zeitschritt k
$\bar{\boldsymbol{z}}$	Hilfsvektor
z	Laufindex

Griechische Buchstaben

Symbol	Beschreibung
$\boldsymbol{\alpha}$	Vektor
$\alpha, \alpha_{\mathrm{I}}$	Grad der PE
$\alpha^{[X]}, \alpha^{[Y]}$	Plattenwinkel in X- bzw. Y-Richtung
α_{t}	Schwellwert
β	Faktor für gewichtete Least-Squares-Schätzung bzw. Schwimmwinkel
$\boldsymbol{\gamma}(\cdot)$	funktionaler Zusammenhang der Spalten der Eingangsmatrix $\boldsymbol{g}(\boldsymbol{x})$
γ	Diskontierungsfaktor
$\boldsymbol{\delta}$	TD-Fehlervektor
δ	relativer Grad/Differenzordnung bzw. TD-Fehler
δ_{LR}	Lenkradwinkel
$\delta_{\mathrm{LR,v}}$	Winkelgeschwindigkeit des Lenkrads
δ_{s}	Lenkwinkel
ϵ	Approximationsfehler
$\bar{\epsilon}$	obere Schranke des Approximationsfehlers
$\boldsymbol{\varepsilon}_1$	beschränktes Fehlersignal

Symbol	Beschreibung
ε_2	transienter Fehler
$\bar{\varepsilon}_\sigma$	obere Schranke von ε_1
$\boldsymbol{\zeta}$	Referenzparametervektor
ζ	Frequenzvariable
η	Lernrate
$\boldsymbol{\theta}$	Actor-Gewicht
ι	Platzhalter
κ	Laufindex
$\lambda_{\mathrm{min},1}$	Eigenwertsignal (minimaler Eigenwert von Ξ_1)
$\lambda_{\mathrm{min},2}$	Eigenwertsignal (minimaler Eigenwert von Ξ_2)
λ_{LM}	Regularisierung des Levenberg-Marquardt-Algorithmus
$\boldsymbol{\mu}(\cdot)$	Regelgesetz
ν	Skalierungsfaktor bzw. Messrauschen
$\boldsymbol{\Xi}_1$	PE-Matrix im Intervall $[t, t+T]$
$\boldsymbol{\Xi}_2$	PE-Matrix im Intervall $[t_0, t]$
$\boldsymbol{\rho}(\cdot)$	Basisfunktionen Referenzapproximation
ρ	Konvergenzrate
$\boldsymbol{\sigma}$	für Adaption relevantes Signal
σ_{s}	skalares Signal
τ	Zeit/Integrationsvariable
$\boldsymbol{\Phi}$	Hilfsmatrix für den LSPI-Algorithmus (Minimierung quadrierter TD-Fehler)
$\bar{\boldsymbol{\Phi}}$	Hilfsmatrix für den LSPI-Algorithmus (mit Fixpunktforderung)
$\boldsymbol{\phi}(\cdot)$	Basisfunktionsvektor
$\boldsymbol{\Psi}_1$	funktionaler Zusammenhang zwischen Zustand und flachem Ausgang sowie dessen Ableitungen
$\boldsymbol{\Psi}_2$	funktionaler Zusammenhang zwischen Stellgröße und flachem Ausgang sowie dessen Ableitungen
Ψ	Menge der zulässigen Regelgesetze
$\boldsymbol{\psi}(\cdot)$	Basisfunktionsvektor
ψ	Gierwinkel
ψ_{r}	Gierrate
Ω	Menge an Frequenzbedingungen
$\boldsymbol{\omega}$	Frequenzvektor
$\omega, \underline{\omega}$	Frequenz
$\omega^{[X]}, \omega^{[Y]}$	Winkelgeschwindigkeit Platte in X- bzw. Y-Richtung

Kalligraphische und andere Symbole

Symbol	Beschreibung
$\mathbf{0}_{n\times m}$	Nullmatrix der Dimension $n \times m$
$\mathcal{B}$	Bellman-Operator
$\mathcal{C}^{\iota}(\mathcal{M})$	Klasse der auf einer Menge $\mathcal{M}$ ι-fach stetig differenzierbaren Funktionen
$\mathcal{D}$	Menge der Dimensionen des Ball-auf-Platte-Systems
$\mathbb{N}$	Menge der natürlichen Zahlen
$\mathcal{N}$	Menge an Spielern/Reglern
$\mathcal{N}(\mu; \sigma^2)$	Normalverteilung mit Mittelwert μ und Standardabweichung σ
$\mathcal{P}$	Frequenztupel
$\mathbb{R}$	Menge der reellen Zahlen
$\mathbb{R}_{\geq 0}$	Menge der nicht-negativen reellen Zahlen
$\mathcal{S}\{\cdot\}$	Transformation eines Signals
$\mathcal{T}$	Datentupel
$\mathcal{X}$	Menge der Systemzustände
$\mathcal{Z}$	Definitionsbereich des Referenzparametervektors $\boldsymbol{\zeta}$

Indizes, Exponenten und Operatoren

Symbol	Beschreibung
$\mathrm{Bild}(\cdot)$	Bild einer Matrix
$\det(\cdot)$	Determinante einer Matrix
$\mathrm{diag}(\cdot)$	Diagonalmatrix
$\mathrm{Dim}(\cdot)$	Dimension einer Matrix
$\mathrm{eig}(\cdot)$	Eigenwerte einer Matrix
$\mathrm{Im}\{\cdot\}$	Imaginärteil
$L_{\boldsymbol{f}}^{j} h(\boldsymbol{x})$	j-te Lie-Ableitung von $h(\boldsymbol{x})$ bezüglich $\boldsymbol{f}(\boldsymbol{x})$
$\mathrm{mat}(\cdot)$	Konvertierung eines Vektors in eine Matrix, wobei $\mathrm{mat}(\mathrm{vec}(\boldsymbol{M}), n, p) = \boldsymbol{M} \in \mathbb{R}^{n\times p}$
mod	Modulo
$\mathrm{Rang}(\cdot)$	Rang einer Matrix
$\mathrm{Re}\{\cdot\}$	Realteil
$\mathrm{sgn}(\cdot)$	Signumfunktion
$\mathrm{std}(\cdot)$	Standardabweichung
$\mathrm{vec}(\cdot)$	Vektorisierung einer Matrix (vertikale Konkatenation der Spalten)
$\mathrm{vecr}(\cdot)$	Vektorisierung einer symmetrischen Matrix $\boldsymbol{M}$, sodass $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{M}\boldsymbol{x} = (\boldsymbol{x} \otimes_{\mathrm{r}} \boldsymbol{x})^{\mathsf{T}}\mathrm{vecr}(\boldsymbol{M})$
$\otimes$	Kronecker-Produkt
$\otimes_{\mathrm{r}}$	reduziertes Kronecker-Produkt mit nicht-redundanten Elementen

Symbol	**Beschreibung**
$\lambda^{+}_{\mathrm{Re}}(\cdot)$	größter Eigenwert einer reellwertigen Matrix
$\lambda_{\min}(\cdot)$	kleinster Eigenwert einer Matrix
$\nabla_{\boldsymbol{x}}$	Gradient bezüglich $\boldsymbol{x}$, sodass $\nabla_{\boldsymbol{x}} V(\boldsymbol{x}) := \begin{bmatrix} \frac{\partial V(\boldsymbol{x})}{\partial x_1} & \dots & \frac{\partial V(\boldsymbol{x})}{\partial x_n} \end{bmatrix}^\mathsf{T}$
∂	partielle Ableitung
$\succ$	$\boldsymbol{M}_1 \succ \boldsymbol{M}_2$: Matrix $\boldsymbol{M}_1 - \boldsymbol{M}_2$ positiv definit
$\succeq$	$\boldsymbol{M}_1 \succeq \boldsymbol{M}_2$: Matrix $\boldsymbol{M}_1 - \boldsymbol{M}_2$ positiv semidefinit
$\forall$	Allquantor
$\exists$	Existenzquantor
$\lVert\cdot\rVert_2$	Spektralnorm einer Matrix bzw. Euklidische Norm eines Vektors
$\lVert\cdot\rVert_\infty$	Maximumsnorm
$\cdot$	Platzhalter
$\left[\cdot\right]_{n_1:n_2}$	Teilvektor: $\left[\boldsymbol{v}\right]_{n_1:n_2} = \begin{bmatrix} v_{n_1} & \dots & v_{n_2} \end{bmatrix}^\mathsf{T}$ mit $n_1, n_2 \in \mathbb{N}_{>0}, n_1 < n_2$
$\hat{\square}$	geschätzte Größen
$\check{\square}$	aus exakter Systeminversion resultierende Größen
$\dot{\square}$	zeitliche Ableitung
$\square^*$	Optimalität bzw. Nash-Lösung
$\square^\star$	Komplexe Konjugation
$\bar{\square}$	Kennzeichnung von Hilfsgrößen bzw. Komplementärmengen
$\widetilde{\square}$	Fehlersignal
$\tilde{\square}$	Kennzeichnung von Größen eines erweiterten Systems bzw. Hilfsgrößen
$\square^\mathsf{T}$	Transposition einer Matrix oder eines Vektors
$\square^{(j)}(t)$	j-te zeitliche Ableitung einer zeitkontinuierlichen Größe
$\square^{(\kappa)}$	um κ Zeitschritte verschobene zeitdiskrete Größe
$\square^{\{k\}}$	Kennzeichnung von Größen aus lokaler Perspektive im Zeitschritt k
$\square^{[l]}$	Kennzeichnung von Größen der l-ten Iteration
$\square^{\langle\cdot\rangle}$	Indizierung zur Kennzeichnung von Abhängigkeiten von einer Größe

1 Einleitung

Lernbasierte, adaptive Optimalregelungsmethoden rücken zunehmend in den Fokus der aktuellen Forschung [WHL17], [SB18], [KHL+12]. Während modellbasierte optimale Regelungsansätze (vgl. beispielsweise [Gee07], [SMC+11], [TBBH10]) nur anwendbar sind, wenn ein Systemmodell vorhanden ist [LV09], [VLV13, S. 1], liegt der Fokus von selbstlernenden Optimalreglern auf Anwendungen, bei denen kein oder nur unvollständiges Systemwissen vorliegt, eine Modellbildung aufwendig ist oder unbekannte Parameter schwierig zu ermitteln sind [HJK20]. Adaptive, lernende Ansätze erlauben somit trotz vorhandener Modellunsicherheiten den Entwurf leistungsfähiger, flexibler Regler [Wer99], [Wer92], [SBW92], [Tao03, S. 10], [HLM03]. Flexible Reglerentwurfsmethoden [CF14], [LLSX14] und damit einhergehende Fertigungsprozesse [MMP+10] versprechen schließlich Produkte und Anwendungen, die an kundenspezifische Wünsche und individuelle Nutzungsanforderungen angepasst werden können. Derartige Anforderungen sind insbesondere im Kontext der Industrie 4.0 verstärkt zu erwarten [Pla19]. Im Vergleich zu nicht-optimierungsbasierten adaptiven Regelungsansätzen (vgl. beispielsweise [ÅW95], [Tao03], [IS96]) steht bei den in der vorliegenden Arbeit betrachteten Methoden die Minimierung eines zugrunde liegenden Gütefunktionals bei der Adaption von Reglergewichten im Zentrum (vgl. [KHL+12], [LVV12], [VLV13, S. 4]). Dabei können gewünschte Designziele, wie beispielsweise Kosten, Präzision, Zeitverhalten, Energieverbrauch, Komfort, Sicherheit und individuelle Bedürfnisse, durch Gütemaße berücksichtigt werden [LVS12], [AM89], [Ber95].

Reglerparameter basierend auf Simulations- oder Messdaten sowie einem Belohnungssignal zu adaptieren, um ein optimales Regelgesetz zu erlernen, entspricht dem Grundprinzip des Reinforcement Learning (RL) [SB18]. Mithilfe von RL ist es in den letzten Jahren gelungen, tiefe neuronale Netze zu trainieren und komplexe Aufgaben zu lösen. So konnten Meister des Brettspiels Go erstmalig durch einen Computer besiegt [SSS+17], [SHM+16], Greifbewegungen für Roboter unter Nutzung von Kameradaten gelernt [KIP+18] und Bewegungen einer robotischen Hand erlernt werden [ABC+20]. Jedoch musste für diese Erfolge ein erheblicher Trainingsaufwand betrieben werden, insbesondere wurden sehr große Mengen an Trainingsdaten benötigt. Beispielsweise spielte der Go-Computer AlphaGo Zero [SSS+17] im Laufe des Trainingsprozesses fast fünf Millionen komplette Partien gegen sich selbst, Kalashnikov et al. [KIP+18] sammelten Trainingsdaten aus 580 000 Greifversuchen über einen Zeitraum von vier Monaten und betrieben dazu gleichzeitig sieben Roboterarme, während Andrychowicz et al. [ABC+20] mithilfe einer Simulationsumgebung künstlich generierte Trainingsdaten nutzten, die, abhängig von der Variabilität der Umgebung, drei bis 100 Jahren an realen Erfahrungsdaten entsprachen.

Aus einer anwendungsorientierten und regelungstechnischen Perspektive sind solche außerordentlichen Datenmengen zum Training in bestimmten Anwendungen jedoch nicht verfügbar oder unpraktikabel [DAMH19], [BBdE10, S. 8]. Beispiele sind Messdaten realer technischer Systeme, Mensch-Maschine-Systeme oder biomedizinische Anwendungen (vgl. [LL04], [HMT+21], [GSK19]). Aus diesem Grund wird in der vorliegenden Arbeit das Konzept des sogenannten Adaptive Dynamic Programming (ADP)[1] [Wer92], [BT96], [MCLS02], [LV09], [LVV12] betrachtet, das eine Kombination aus adaptiver und optimaler Regelung darstellt [KHL+12] und Mechanismen des Reinforcement Learning nutzt. Die Einbeziehung gegebenenfalls vorhandenen Vorwissens über das System und die zugrunde liegende Problemstellung, wie beispielsweise die Systemordnung oder die Struktur der gesuchten Lösung, kann hierbei die Komplexität des Trainingsvorgangs und somit die Menge der benötigten Trainingsdaten entscheidend reduzieren (vgl. [Gör17], [BBdE10, Abschnitt 3.7.3 und Abschnitt 5.4]). ADP stellt zudem im Gegensatz zum klassischen, indirekten Vorgehen, zunächst ein Systemmodell zu identifizieren und anschließend eine modellbasierte Optimierung vorzunehmen, einen ganzheitlichen (vgl. [SB18, S. 3]), direkten Ansatz dar [LV09, S. 41], [SBW92], [JJ12], [FWS+18].

Dieser regelungstechnischen und anwendungsorientierten Perspektive folgend werden in der vorliegenden Arbeit zwei zentrale und bislang nur unzureichend gelöste Probleme ADP-basierter Regelungsansätze behandelt:

1. ADP-basierte Solltrajektorienfolgeregler und
2. Konvergenzbedingungen für eine erfolgreiche Adaption.

Die Relevanz der ersten Problemstellung resultiert dabei aus der Anforderung vieler technischer Anwendungen, wie beispielsweise Fahrzeugen, Robotern oder verfahrenstechnischen Anlagen, dass Systemgrößen einer vorgegebenen Referenz- bzw. Solltrajektorie folgen sollen (vgl. z. B. [BK18], [FOSH14], [SCN+04], [LLHW16], [MBTL12], [DCP96]). Viele Literaturbeiträge zu ADP-basierten Methoden beschränken sich jedoch entweder auf den Fall, den Systemzustand optimal bezüglich eines Gütemaßes zu null zu regeln, oder betrachten Referenztrajektorien unter sehr einschränkenden Annahmen. In Anwendungen, wie beispielsweise der Robotik oder dem hochautomatisierten oder autonomen Fahren, die im Allgemeinen eine möglichst flexible und zur Laufzeit veränderbare Solltrajektorienvorgabe erfordern (vgl. [van97]), sind bestehende ADP-Ansätze daher ungeeignet. Eine wesentliche Herausforderung beim Entwurf von ADP-Zustandstrajektorienfolgereglern ist eine Repräsentation der Solltrajektorie, die in den ADP-Formalismus integriert werden kann. Eine hierfür geeignete Darstellung des Sollzustandsverlaufs, die einerseits eine flexible Solltrajektorienvorgabe ermöglicht und andererseits mit einem vertretbaren Lernaufwand verbunden ist, ist dabei nicht trivial, speziell aus einer anwendungsorientierten Perspektive jedoch unverzichtbar. Bislang existieren

[1] Weitere Bezeichnungen sind *Approximate Dynamic Programming* [Wer13], *Neuro-Dynamic Programming* [BT96], *Heuristic Dynamic Programming* [Wer77] oder *Incremental Dynamic Programming* [Wat89]. In der Literatur werden die Begriffe ADP und RL gelegentlich synonym verwendet [Gos09]. Eine präzise Abgrenzung ist im Allgemeinen nicht möglich [Wer13], die Bezeichnung ADP deutet jedoch meist auf einen eher regelungstechnisch orientierten Zweig des RL hin (vgl. [Gör17], [BBT+18], [LV09]).

somit weder eine allgemeine theoretische Definition, unter welchen Bedingungen eine Solltrajektoriendarstellung kompatibel für die Verwendung im ADP-Kontext ist, noch geeignete Ansätze ADP-basierter Solltrajektorienfolgeregler, welche die Einbeziehung einer von außen vorgebbaren Repräsentation variabler Solltrajektorienverläufe ermöglichen. Die vorliegende Arbeit schließt diese Lücke. Neben der theoretischen Analyse, Simulation und Diskussion der präsentierten Ansätze wird zudem die Anwendbarkeit von ADP-basierten Solltrajektorienfolgereglern anhand zweier realer Anwendungsbeispiele untersucht.

Ein zweiter wesentlicher Aspekt ADP-basierter Regler ist die Frage, unter welchen Bedingungen die Konvergenz des Lernprozesses gewährleistet ist. Diese Frage hängt eng mit dem Begriff der Exploration aus dem Bereich des RL zusammen. Dabei muss stets ein Kompromiss zwischen dem Ausprobieren neuer Stellgrößen und Zustände, die möglicherweise bislang unbekannte, bessere Optionen zutage bringen, und dem Ausnutzen bisheriger Erfahrungen eingegangen werden (Exploration-Exploitation-Dilemma) [SB18, S. 3]. Anschaulich betrachtet müssen die Trainingsdaten kausale Zusammenhänge, wie das Systemverhalten, den Einfluss potenzieller weiterer Akteure und die resultierenden Kosten, angemessen abbilden. Für eine erfolgreiche Konvergenz von ADP-Reglern ist somit eine ausreichende Anregung des Systems bzw. der für die Adaption relevanten Signale erforderlich. Speziell im Kontext des ADP existieren bislang jedoch kaum theoretische Erkenntnisse, wie eine geeignete Anregung entworfen werden kann [JKBD15] und welche Bedingungen die Systemzustände erfüllen müssen, um Konvergenz des Adaptionsprozesses zu erzielen. Der zweite zentrale Beitrag der vorliegenden Arbeit ist daher durch hinreichende Bedingungen an den Systemzustand gegeben, die gewährleisten, dass die für die Konvergenz benötigte Anregung des Systems erfüllt ist. Die theoretischen Ergebnisse werden anhand von Simulationen analysiert und diskutiert.

Gliederung der Arbeit

Die vorliegende Arbeit ist wie folgt gegliedert: **Kapitel 2** gibt einen Überblick über den relevanten Stand der Wissenschaft und führt benötigte Notationen und Begriffe ein. Dabei werden Forschungslücken herausgearbeitet, Forschungsfragen konkretisiert und ein Überblick über die Beiträge der vorliegenden Arbeit gegeben. Die Kapitel 3–6 bilden anschließend den Kern der Arbeit. Eine schematische Übersicht dieser Hauptinhalte ist in Abbildung 1.1 dargestellt.

Dabei werden in **Kapitel 3** und **Kapitel 4** flexible Referenztrajektoriendarstellungen, die mit dem ADP-Formalismus kompatibel sind, vorgestellt. In zeitdiskreter Formulierung (Kapitel 3) werden einerseits parametrierte Solltrajektorienverläufe für den Entwurf eines ADP-basierten Trajektorienfolgereglers entwickelt, andererseits wird eine Methode zur direkten Verwendung der Sollzustände auf einem endlichen Vorausschauhorizont vorgestellt. In zeitkontinuierlicher Darstellung (Kapitel 4) werden ebenfalls parametrierte, ADP-kompatible Sollzustandsverläufe präsentiert und in einen ADP-Ansatz integriert. Die Eigenschaften der unterschiedlichen Methoden werden analysiert und diskutiert. Zudem werden jeweils Simulationsergebnisse präsentiert.

Kapitel 5 liefert formale theoretische Aussagen zur Anregung ADP-basierter Regelungsansätze. Für die verallgemeinerte Problemstellung eines zeitkontinuierlichen eingangsaffinen Nicht-Nullsummen-Differenzialspiels werden hinreichende Bedingungen an den Systemzustand präsentiert, welche die Erfüllung der Anregungsbedingung und somit Konvergenz der betrachteten ADP-Methode garantieren. Illustrative Simulationsergebnisse werden präsentiert und diskutiert.

In **Kapitel 6** werden adaptive optimale Solltrajektorienfolgeregler anhand zweier realer Anwendungsbeispiele präsentiert. Zunächst wird ein adaptiver Geschwindigkeitsregler, der einem vorgegebenen Geschwindigkeitsprofil möglichst kostenoptimal folgen soll, in einem realen Fahrzeug angewandt. Dieser wird online, d. h. während sich das Fahrzeug auf der Teststrecke befindet, adaptiert. Schließlich wird als zweites reales Anwendungsbeispiel ein adaptiver optimaler Trajektorienfolgeregler für ein Ball-auf-Platte-System vorgestellt, der anhand von aufgezeichneten Messdaten und ohne Verwendung eines konkreten Systemmodells erlernt wird.

Kapitel 7 fasst schließlich die Haupterkenntnisse der Arbeit zusammen und bewertet diese.

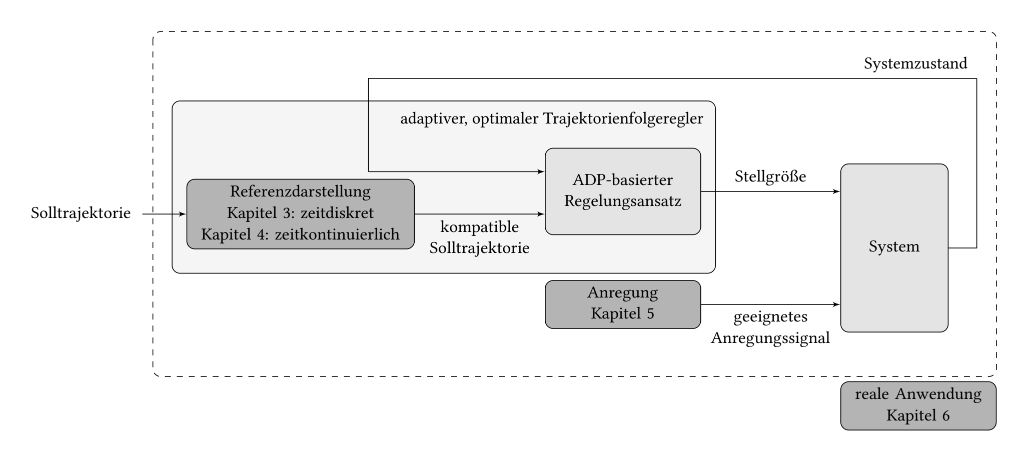


Abbildung 1.1: Schematische Übersicht der Hauptbeiträge der vorliegenden Arbeit.

2 Stand der Wissenschaft und Forschungslücke

In Abschnitt 2.1 dieses Kapitels werden zunächst zentrale Grundgleichungen und Grundbegriffe des ADP eingeführt, um eine Einordnung und Diskussion bestehender Methoden zu ermöglichen. Anschließend wird in Abschnitt 2.2 der Stand der Wissenschaft hinsichtlich bestehender ADP-basierter Solltrajektorienfolgeregelungsmethoden diskutiert. Weiterhin wird in Abschnitt 2.3 beleuchtet, wie die Anregung von Systemzuständen im ADP-Kontext in der Literatur bislang erfolgt. Nachdem die Forschungslücken herausgearbeitet wurden, werden in Abschnitt 2.4 die wissenschaftlichen Fragestellungen der Arbeit präzisiert und eine Übersicht über die Beiträge präsentiert.

2.1 Einführung der Notation und Grundbegriffe des ADP

ADP-basierte Regelungsansätze, die mit RL-Methoden und gemessenen Systemtrajektorien Optimalregelungsprobleme lösen sollen (vgl. [LV09, S. 39]), stellen ein vergleichsweise junges Gebiet der Regelungstechnik dar (vgl. [VLV13, S. 2], [SB18, Abschnitt 1.7], [Wer13]). Zwar postulieren Mendel und McLaren [MM70] bereits 1970 erste trial-and-error-basierte Regelungsansätze, die Ideen des RL aufgreifen, Werbos [Wer77] beschreibt 1977 das Konzept des *Heuristic Dynamic Programming* und Barto et al. [BSA83] gelingen durch Diskretisierung des Zustandsraums[2] simulative Trainingserfolge, dennoch beginnt die eigentliche Entwicklung moderner ADP-Ansätze erst mit der Arbeit von Watkins [Wat89] (vgl. [SB18, S. 14]). Eine ausführliche Übersicht über die Ursprünge und Grundlagen von RL- und ADP-Methoden ist beispielsweise in [LVV12], [SB18, Kapitel 1], [LV09], [WZL09], [LWW^{+}17, Kapitel 1] und [Wer13] gegeben.

ADP- und RL-basierte Ansätze lassen sich in zwei wesentliche Klassen unterteilen. Neben Monte-Carlo-Methoden [SB18, Kapitel 5] spielen insbesondere auf dem sogenannten *Temporal-Difference-Fehler* (TD-Fehler) [Sut88], [SB18, Kapitel 6] beruhende Methoden eine Rolle. Letztere nutzen einen skalaren Prädiktionsfehler, der auf der Bellman-Gleichung bzw. der Hamilton-Jacobi-Bellman-Gleichung basiert, zur Adaption. Monte-Carlo-Methoden betrachten üblicherweise episodische Aufgaben (vgl. [SB18, S. 91]) mit klar definierten Endzuständen, welche viele Male wiederholt werden. Hierbei müssen Lernalgorithmen stets das Ende einer Episode abwarten, wohingegen Algorithmen, die den TD-Fehler nutzen, den Vorteil aufweisen, mit

[2] Dieser unter dem Begriff des *tile coding* bekannte Mechanismus kann jedoch zahlreiche theoretische und praktische Probleme verursachen (vgl. [van12, Abschnitt 2.1.2]).

jedem Zeitschritt ein Datentupel zu erhalten und unabhängig eines episodischen Charakters Adaptionen vornehmen zu können [LVV12, S. 87 f.], [SB18, Abschnitt 6.2], [VLV13, S. 29 f.]. Ein weiteres Argument, Anpassungen auf Basis eines skalaren Vorhersagefehlers, wie dem TD-Fehler, vorzunehmen, ist, dass dieses Grundprinzip des RL über den rein technischen Ursprung hinausgeht[3]. Im Folgenden wird der Fokus dieser Arbeit auf die Klasse der TD-Methoden gelegt.

Um den für die vorliegende Arbeit relevanten Stand der Wissenschaft angemessen einordnen zu können, werden in den nachfolgenden Abschnitten zunächst im ADP-Kontext wichtige Grundgleichungen in zeitdiskreter und zeitkontinuierlicher Darstellung eingeführt. Anschließend wird die Notwendigkeit der Verwendung von Funktionsapproximatoren bei wertkontinuierlichen Zustands- und Stellgrößenräumen diskutiert. Hierbei dient ein Funktionsapproximator beispielsweise der Beschreibung des funktionalen Zusammenhangs zwischen dem Systemzustand und den daraus resultierenden Kosten. Schließlich werden grundlegende ADP-Lösungsansätze vorgestellt.

2.1.1 Zeitdiskrete ADP-Grundgleichungen

In zeitdiskreter Darstellung werde zunächst ein System

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{g}(\boldsymbol{x}_k)\boldsymbol{u}_k, \tag{2.1}$$

$\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{R}^p$, $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$, $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^{n \times p}$, $\boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{0}$, mit dem Zeitindex k und dem zu minimierenden[4] Gütefunktional

$$J(\boldsymbol{x}_0, \boldsymbol{\mu}) = \sum_{\kappa=0}^{\infty} \gamma^\kappa \big(q(\boldsymbol{x}_\kappa) + \boldsymbol{\mu}(\boldsymbol{x}_\kappa)^\intercal \boldsymbol{R} \boldsymbol{\mu}(\boldsymbol{x}_\kappa)\big) =: \sum_{\kappa=0}^{\infty} \gamma^\kappa r(\boldsymbol{x}_\kappa, \boldsymbol{\mu}(\boldsymbol{x}_\kappa)) \tag{2.2}$$

betrachtet (vgl. beispielsweise [ATLAK08], [LV09], [LVS12, Kapitel 11.5] und [WZL09]). Hierbei stellt $\boldsymbol{\mu} : \mathbb{R}^n \to \mathbb{R}^p$ eine Zustandsrückführung dar. Weiterhin sei $q : \mathbb{R}^n \to \mathbb{R}$ eine positiv definite Funktion (vgl. [NA05, S. 53]) und $\boldsymbol{R}$ symmetrisch und positiv definit. Zudem sei das

[3] So können Parallelen zu verhaltenspsychologischen Modellen und der Neurobiologie gezogen werden. Bentham [Ben89] definiert bereits 1789 das Nützlichkeitsprinzip, nach welchem Handlungen bevorzugt werden, die den Gesamtnutzen vergrößern. 1911 spricht Thorndike [Tho11], das Lernverhalten von Tieren untersuchend, vom „Law of Effect". Demnach werden Aktionen, die zu einer Belohnung geführt haben, in künftigen, ähnlichen Situationen häufiger wiederholt als mögliche Alternativen. Fortgesetzt wird dieser behavioristische Ansatz durch Watsons Grundsatz, Verhalten lasse sich durch Reiz und Reaktion beschreiben [Wat13]. Zudem findet laut Rescorla und Wagner [RW72], die auf Pavlovs klassischer Konditionierung [Pav27] aufbauen, gerade dann ein Lernprozess statt, wenn sich Prädiktion und tatsächliche Beobachtung unterscheiden. Somit ist ein Vorhersagefehler treibende Kraft des Lernens. Weiterhin lassen sich Parallelen zwischen dem TD-Learning und Lernvorgängen im menschlichen Gehirn, die auf dem Neurotransmitter Dopamin beruhen, ziehen. So lässt sich die Dopaminaktivität im Mittelhirn, ähnlich wie der TD-Fehler, als skalares Prädiktionsfehlersignal interpretieren, welches Lernvorgänge auf Neuronenebene beeinflusst [SDM97], [Gli11], [Niv09], [HDB94]. Letztlich lässt sich durch ADP-Methoden auch menschliches Verhalten beim Erlernen sensomotorischer Bewegungen nachbilden [JJ14a], [Bia17].

[4] In dieser Arbeit wird die in der optimalen Regelung übliche Konvention der Minimierung von Kosten verwendet, während im Bereich des RL eine Maximierung von Belohnungen üblich ist (vgl. [LVV12]).

System stabilisierbar auf der kompakten Menge $\mathcal{X} \subset \mathbb{R}^n$, die den Ursprung enthält. Durch den sogenannten Diskontierungsfaktor γ mit $0 < \gamma \leq 1$ lässt sich parametrieren, ob die Einschrittkosten $r(\cdot)$ zu jedem Zeitschritt gleich gewichtet werden ($\gamma = 1$), oder inwiefern weiter in der Zukunft liegende Kosten schwächer ins Gewicht fallen ($\gamma < 1$). Weiterhin stellt der Diskontierungsfaktor ein Werkzeug dar, um auch bei nicht-verschwindenden Einschrittkosten $r(\cdot)$ einen endlichen Wert des Gütefunktionals erhalten zu können (vgl. Abschnitt 2.2).

Eine wichtige Größe, die eng mit dem Gütefunktional $J(\boldsymbol{x}_0, \boldsymbol{\mu})$ in (2.2) verknüpft ist, stellt die sogenannte Zustands-Nutzenfunktion[5], die auch als *Value Function* bezeichnet wird, dar. Die Value Function ist für zulässige[6], d. h. stabilisierende, stetige[7] Regler $\boldsymbol{\mu}(\boldsymbol{x}_k) \in \mathcal{C}^0(\mathcal{X})$, die zudem zu einem endlichen Gütefunktional führen (vgl. [ATLAK08, Definition 2]), durch

$$V^{\boldsymbol{\mu}}(\boldsymbol{x}_k) = \sum_{\kappa=k}^{\infty} \gamma^{\kappa-k} r(\boldsymbol{x}_\kappa, \boldsymbol{\mu}(\boldsymbol{x}_\kappa)) = r(\boldsymbol{x}_k, \boldsymbol{\mu}(\boldsymbol{x}_k)) + \gamma V^{\boldsymbol{\mu}}(\boldsymbol{x}_{k+1}), \tag{2.3a}$$

$$V^{\boldsymbol{\mu}}(\boldsymbol{0}) = 0, \tag{2.3b}$$

gegeben. Die durch (2.3b) beschriebene Anfangsbedingung folgt aufgrund von $\boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{0}$ und $q(\boldsymbol{x}_k)$ positiv definit. In Übereinstimmung mit Bellmans Optimalitätsprinzip [Bel57a] resultiert für

$$\boldsymbol{\mu}^*(\boldsymbol{x}_k) = \arg\min_{\boldsymbol{\mu}} V^{\boldsymbol{\mu}}(\boldsymbol{x}_k) \tag{2.4}$$

aus (2.3a) die Bellman-Gleichung

$$V^*(\boldsymbol{x}_k) \coloneqq V^{\boldsymbol{\mu}^*}(\boldsymbol{x}_k) = \min_{\boldsymbol{\mu}(\boldsymbol{x}_k)} r(\boldsymbol{x}_k, \boldsymbol{\mu}(\boldsymbol{x}_k)) + \gamma V^*(\boldsymbol{x}_{k+1}). \tag{2.5}$$

Das optimale Regelgesetz ergibt sich zu

$$\boldsymbol{\mu}^*(\boldsymbol{x}_k) = -\frac{\gamma}{2} \boldsymbol{R}^{-1} \boldsymbol{g}^\intercal(\boldsymbol{x}_k) \nabla_{\boldsymbol{x}_{k+1}} V^*(\boldsymbol{x}_{k+1}) \tag{2.6}$$

(vgl. [LVV12]). Neben der Value Function $V^{\boldsymbol{\mu}}(\boldsymbol{x}_k)$ spielt die sogenannte Zustands-Aktions-Nutzenfunktion[8], die in Anlehnung an klassisches Q-Learning [Wat89], [WD92] auch als *Q-Function*[9] bezeichnet wird, eine wichtige Rolle in der ADP-Literatur (vgl. beispielsweise [KLM+14], [LLW+17], [LLHW16], [ATLAK07], [LVV12]). Die Q-Function ist durch

$$\begin{aligned} Q^{\boldsymbol{\mu}}(\boldsymbol{x}_k, \boldsymbol{u}_k) &\coloneqq r(\boldsymbol{x}_k, \boldsymbol{u}_k) + \sum_{\kappa=k+1}^{\infty} \gamma^{\kappa-k} r(\boldsymbol{x}_\kappa, \boldsymbol{\mu}(\boldsymbol{x}_\kappa)) \\ &= r(\boldsymbol{x}_k, \boldsymbol{u}_k) + \gamma Q^{\boldsymbol{\mu}}(\boldsymbol{x}_{k+1}, \boldsymbol{\mu}(\boldsymbol{x}_{k+1})) \\ &= r(\boldsymbol{x}_k, \boldsymbol{u}_k) + \gamma V^{\boldsymbol{\mu}}(\boldsymbol{x}_{k+1}) \end{aligned} \tag{2.7}$$

5 (engl.): *state value function.* Per Definition gilt $V^{\boldsymbol{\mu}}(\boldsymbol{x}_0) = J(\boldsymbol{x}_0, \boldsymbol{\mu})$.
6 (engl.): *admissible.*
7 Die Notation $\mathcal{C}^\iota(\mathcal{X})$ kennzeichnet die Klasse der auf der Menge $\mathcal{X}$ ι-fach stetig differenzierbaren Funktionen.
8 (engl.): *state-action value function.*
9 Diese Bezeichnung geht auf den Begriff *quality function* zurück [LVV12].

definiert. Sie beschreibt die (diskontierten) Langzeitkosten, wenn sich das System im Zustand $\boldsymbol{x}_k$ befindet, im Zeitschritt k die Stellgröße $\boldsymbol{u}_k$ und in allen weiteren Zeitschritten das Regelgesetz $\boldsymbol{\mu}(\boldsymbol{x}_\kappa)$ angewandt wird. Mit $Q^*(\boldsymbol{x}_k, \boldsymbol{u}_k) \coloneqq Q^{\boldsymbol{\mu}^*}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ ergibt sich das optimale Regelgesetz zu

$$\boldsymbol{\mu}^*(\boldsymbol{x}_k) = \boldsymbol{u}^*(\boldsymbol{x}_k) = \arg\min_{\boldsymbol{u}_k} Q^*(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{2.8}$$

[LLW+17], [BBdE10, S. 18], [LVV12], zudem gilt

$$V^*(\boldsymbol{x}_k) = \min_{\boldsymbol{u}_k} Q^*(\boldsymbol{x}_k, \boldsymbol{u}_k). \tag{2.9}$$

Ein wesentlicher Unterschied von (2.8) im Vergleich zu (2.6) ist, dass die Eingangsdynamik $\boldsymbol{g}(\boldsymbol{x}_k)$ darin nicht präsent ist. Dies ist auf die explizite Abhängigkeit der Q-Function von $\boldsymbol{u}_k$ zurückzuführen (vgl. [LVS12, S. 502]). Weiterhin hängt (2.8) nicht von $\boldsymbol{x}_{k+1}$, und somit implizit von der Stellgröße $\boldsymbol{u}_k$ (vgl. [LVS12, S. 497]), sondern ausschließlich von $\boldsymbol{x}_k$ ab.

2.1.2 Zeitkontinuierliche ADP-Grundgleichungen

Basierend auf [VL09], [VL10], [LVV12], [WHL17] und [LVS12, Kapitel 10.1] wird ein eingangsaffines System mit der Dynamik

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{g}(\boldsymbol{x}(t))\boldsymbol{u}(t), \tag{2.10a}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{2.10b}$$

$\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u} \in \mathbb{R}^p$, $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$, $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^{n \times p}$, $\boldsymbol{f}$ und $\boldsymbol{g}$ Lipschitz-stetig auf der kompakten Menge $\mathcal{X} \subset \mathbb{R}^n$, die den Ursprung enthält, $\boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{0}$, (2.10) stabilisierbar auf $\mathcal{X}$, und ein zu minimierendes Gütefunktional der Form

$$J(\boldsymbol{x}_0, \boldsymbol{\mu}) = \int_0^\infty q(\boldsymbol{x}) + \boldsymbol{\mu}^\intercal(\boldsymbol{x})\boldsymbol{R}\boldsymbol{\mu}(\boldsymbol{x})\,\mathrm{d}\tau =: \int_0^\infty r(\boldsymbol{x}, \boldsymbol{\mu})\,\mathrm{d}\tau \tag{2.11}$$

betrachtet[10]. Dabei sei $q : \mathbb{R}^n \to \mathbb{R}$ abermals eine positiv definite Funktion und $\boldsymbol{\mu} : \mathbb{R}^n \to \mathbb{R}^p$ eine Zustandsrückführung. Zudem gelte $\boldsymbol{R} = \boldsymbol{R}^\intercal \succ \boldsymbol{0}$. Diese Form des Systems (2.10) und Gütefunktionals (2.11) ist in der ADP-Literatur weitverbreitet und schließt insbesondere für $\boldsymbol{f}(\boldsymbol{x}(t)) = \boldsymbol{A}\boldsymbol{x}(t)$ und $\boldsymbol{g}(\boldsymbol{x}(t)) = \boldsymbol{B}$ sowie $q(\boldsymbol{x}) = \boldsymbol{x}^\intercal\boldsymbol{Q}\boldsymbol{x}$ mit $\boldsymbol{Q} \succeq \boldsymbol{0}$ die Problemklasse der linear-quadratischen (LQ-)Optimierungsprobleme ein.

[10] Ähnlich wie bei der zuvor in Abschnitt 2.1.1 vorgestellten zeitdiskreten Formulierung lassen sich auch die zeitkontinuierlichen Gleichungen auf ein diskontiertes Gütemaß verallgemeinern. Mit $\gamma_\mathrm{c} \in \mathbb{R}_{\geq 0}$ wird die später in (2.12) definierte Value Function dann zu $V^{\boldsymbol{\mu}}(\boldsymbol{x}(t)) = \int_t^\infty \mathrm{e}^{-\gamma_\mathrm{c}(\tau - t)} r(\boldsymbol{x}, \boldsymbol{\mu})\,\mathrm{d}\tau = \int_t^{t+T_\mathrm{IRL}} \mathrm{e}^{-\gamma_\mathrm{c}(\tau - t)} r(\boldsymbol{x}, \boldsymbol{\mu})\,\mathrm{d}\tau + \mathrm{e}^{-\gamma_\mathrm{c}(T_\mathrm{IRL} - t)} V^{\boldsymbol{\mu}}(\boldsymbol{x}(t + T_\mathrm{IRL}))$ und die Lyapunov-Gleichung nach (2.13) zu $0 = r(\boldsymbol{x}, \boldsymbol{\mu}) + (\nabla_{\boldsymbol{x}} V^{\boldsymbol{\mu}}(\boldsymbol{x}))^\intercal (\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{\mu}(\boldsymbol{x})) - \gamma_\mathrm{c} V^{\boldsymbol{\mu}}(\boldsymbol{x})$, $V^{\boldsymbol{\mu}}(\boldsymbol{0}) = 0$ (vgl. [Doy00]). Aus Gründen der Übersichtlichkeit ist in Abschnitt 2.1.2 jedoch der Fall $\gamma_\mathrm{c} = 0$ dargestellt.

Die Value Function wird für zulässige Regelgesetze[11] [BSW97, Definition 1] durch

$$V^{\boldsymbol{\mu}}(\boldsymbol{x}) := V^{\boldsymbol{\mu}}(\boldsymbol{x}(t)) = \int_t^{\infty} r(\boldsymbol{x}, \boldsymbol{\mu})\,\mathrm{d}\tau \tag{2.12}$$

definiert [VPAKL09], [VL09]. Somit sind durch die Value Function $V^{\boldsymbol{\mu}}(\boldsymbol{x}(t))$ die Gesamtkosten gegeben, die entstehen, wenn sich das System zum Zeitpunkt t im Zustand $\boldsymbol{x}(t)$ befindet und das Regelgesetz $\boldsymbol{\mu}(\boldsymbol{x})$ verwendet wird.

Die infinitesimale Version von $V^{\boldsymbol{\mu}}(\boldsymbol{x})$ basierend auf (2.12) ist durch die sogenannte Lyapunov-Gleichung (vgl. [BSW97], [VL10])

$$0 = r(\boldsymbol{x}, \boldsymbol{\mu}) + \left(\nabla_{\boldsymbol{x}} V^{\boldsymbol{\mu}}(\boldsymbol{x})\right)^{\mathsf{T}} \left(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{\mu}(\boldsymbol{x})\right), \quad V^{\boldsymbol{\mu}}(\boldsymbol{0}) = 0, \tag{2.13}$$

gegeben[12]. Die mit dem Optimierungsproblem assoziierte Hamilton-Funktion[13] lautet

$$H(\boldsymbol{x}, \nabla_{\boldsymbol{x}} V(\boldsymbol{x}), \boldsymbol{\mu}(\boldsymbol{x})) = r(\boldsymbol{x}, \boldsymbol{\mu}) + \left(\nabla_{\boldsymbol{x}} V(\boldsymbol{x})\right)^{\mathsf{T}} \left(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{\mu}(\boldsymbol{x})\right) \tag{2.14}$$

(vgl. [LLW14], [WHL17]). Die optimale Value Function

$$V^*(\boldsymbol{x}) =: V^{\boldsymbol{\mu}^*}(\boldsymbol{x}) = \min_{\boldsymbol{\mu}} V^{\boldsymbol{\mu}}(\boldsymbol{x}) \tag{2.15}$$

erfüllt unter Annahme der Existenz des Minimums die Hamilton-Jacobi-Bellman-Gleichung (HJB-Gleichung)

$$0 = \min_{\boldsymbol{\mu}} H(\boldsymbol{x}, \nabla_{\boldsymbol{x}} V^*(\boldsymbol{x}), \boldsymbol{\mu}(\boldsymbol{x})). \tag{2.16}$$

Diese entspricht der Lyapunov-Gleichung (2.13) für die optimale Value Function $V^*(\boldsymbol{x})$ und das optimale Regelgesetz $\boldsymbol{\mu}^*$. Zudem gilt nach (2.13) für jedes zulässige Regelgesetz $\boldsymbol{\mu}(\boldsymbol{x})$ mit der zugehörigen Value Function $V^{\boldsymbol{\mu}}(\boldsymbol{x})$ mit $V^{\boldsymbol{\mu}}(\boldsymbol{0}) = 0$

$$0 = H(\boldsymbol{x}, \nabla_{\boldsymbol{x}} V^{\boldsymbol{\mu}}(\boldsymbol{x}), \boldsymbol{\mu}(\boldsymbol{x})). \tag{2.17}$$

Unter der Annahme, dass das Minimum auf der rechten Seite von (2.16) existiert und eindeutig ist, ergibt sich das optimale Regelgesetz zu

$$\boldsymbol{\mu}^*(\boldsymbol{x}) = \arg\min_{\boldsymbol{\mu}(\boldsymbol{x})} H(\boldsymbol{x}, \nabla_{\boldsymbol{x}} V^*(\boldsymbol{x}), \boldsymbol{\mu}(\boldsymbol{x})) = -\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathsf{T}}(\boldsymbol{x})\nabla_{\boldsymbol{x}} V^*(\boldsymbol{x}) \tag{2.18}$$

---

[11] (engl.): *admissible policies*. Diese sind durch $\boldsymbol{\mu}(\boldsymbol{0}) = \boldsymbol{0}$, $\boldsymbol{\mu}(\boldsymbol{x})$ stetig und stabilisierend auf $\mathcal{X}$ und $V^{\boldsymbol{\mu}}(\boldsymbol{x})$ endlich definiert. Letzteres wird hier zusätzlich zu einem stabilisierenden Regelgesetz gefordert, da ein stabilisierender Regler allein nicht gewährleistet, dass das Kostenintegral endlich bleibt [BSW97, S. 2162].

[12] Da im Fall des hier betrachteten unendlichen Optimierungshorizonts keine explizite Abhängigkeit der Value Function von der Zeit existiert, gilt $\frac{\partial V^{\boldsymbol{\mu}}(\boldsymbol{x})}{\partial t} = 0$. Zudem wird angenommen, dass $V^{\boldsymbol{\mu}}(\boldsymbol{x})$ stetig differenzierbar ist [VL10], d. h. $V^{\boldsymbol{\mu}}(\boldsymbol{x}) \in \mathcal{C}^1(\mathcal{X})$.

[13] Hierbei ist $H(\cdot)$ als Funktion mit beliebigem $V(\boldsymbol{x})$ und $\boldsymbol{\mu}(\boldsymbol{x})$ zu verstehen und es muss im Allgemeinen nicht $V(\boldsymbol{x}) = V^{\boldsymbol{\mu}}(\boldsymbol{x})$ gelten.

(vgl. [VL09], [VL10]). Das Einsetzen von (2.18) in (2.16) und die Verwendung von (2.11) liefern die HJB-Gleichung bezüglich $\nabla_{\boldsymbol{x}} V^*$ (vgl. [VL10]):

$$0 = q(\boldsymbol{x}) + \nabla_{\boldsymbol{x}} V^{*\mathsf{T}} \boldsymbol{f}(\boldsymbol{x}) - \frac{1}{4} \nabla_{\boldsymbol{x}} V^{*\mathsf{T}}(\boldsymbol{x}) \boldsymbol{g}(\boldsymbol{x}) \boldsymbol{R}^{-1} \boldsymbol{g}^{\mathsf{T}}(\boldsymbol{x}) \nabla_{\boldsymbol{x}} V^*(\boldsymbol{x}), \quad V^*(\boldsymbol{x}) = 0. \quad (2.19)$$

Falls es nun gelingt, V^* zu finden, sodass (2.19) erfüllt ist, ergibt sich nach (2.18) das optimale Regelgesetz $\boldsymbol{\mu}^*(\boldsymbol{x})$.

Eine Alternative zur direkten Verwendung der Lyapunov-Gleichung (2.13) ergibt sich durch Umformulierung von (2.12) zu

$$V^{\boldsymbol{\mu}}(\boldsymbol{x}) := V^{\boldsymbol{\mu}}(\boldsymbol{x}(t)) = \int_t^{\infty} r(\boldsymbol{x}, \boldsymbol{\mu}) \, \mathrm{d}\tau \quad (2.20a)$$

$$= \int_t^{t+T_{\mathrm{IRL}}} r(\boldsymbol{x}, \boldsymbol{\mu}) \, \mathrm{d}\tau + V^{\boldsymbol{\mu}}(\boldsymbol{x}(t + T_{\mathrm{IRL}})) \quad (2.20b)$$

(vgl. [VPAKL09], [VL09]), wobei $T_{\mathrm{IRL}} > 0$ eine beliebige Intervalllänge darstellt. Gleichung (2.20b) spielt eine zentrale Rolle für ADP-Methoden, die sich unter dem Begriff des *Integral Reinforcement Learning* (IRL) zusammenfassen lassen (vgl. beispielsweise [LVV12], [SLW17], [JJ14c], [BJ16a], [VPAKL09], [VL09], [ML14b], [LS17], [LPC15]). Ein IRL-basierter Ansatz [BJ16a] wird in Kapitel 4 der vorliegenden Arbeit genutzt, während die Lyapunov-Gleichung (2.13) in Kapitel 5 Verwendung findet.

Das Lösen der HJB-Gleichung (2.16) bzw. (2.19) ist im Allgemeinen schwierig [BSW97], [VL09], [VL10], zudem wird die vollständige Kenntnis der Systemdynamik in Form von $\boldsymbol{f}(\boldsymbol{x})$ und $\boldsymbol{g}(\boldsymbol{x})$ benötigt. Ebenso lässt sich aus (2.20b) nicht unmittelbar auf V^* schließen. An dieser Stelle setzen ADP-Methoden an, um basierend auf gemessenen Zustands- und Stellgrößentrajektorien sowie dem Kostensignal $r(\cdot)$ Regelgesetze im Sinne des Gütemaßes J nach (2.11) zu adaptieren. Die wichtigsten Lösungsansätze hierzu werden in Abschnitt 2.1.4 überblicksartig vorgestellt.

2.1.3 Funktionsapproximatoren

Die in dieser Arbeit betrachteten regelungstechnischen Problemstellungen weisen wertkontinuierliche Zustands- und Stellgrößenräume auf. Daher sind klassische RL-Algorithmen ungeeignet, die Markov-Entscheidungsprozesse mit endlichen Zustands- und Stellgrößenräumen betrachten [KLM96] und mit tabellarischen Darstellungen der Value Function V bzw. Q-Function Q arbeiten (z. B. Q-Learning [Wat89], klassisches Temporal-Difference Learning [Sut88], tabellarische Policy-Iteration- [SB18, Kapitel 4.3] oder Value-Iteration-Algorithmen [SB18, Kapitel 4.4]). Insbesondere bei wertkontinuierlichen Zustands- und Stellgrößenräumen ist daher die Verwendung von geeigneten Funktionsapproximatoren entscheidend (vgl. beispielsweise [BBdE10, Kapitel 3.2], [LVV12, S. 90], [van12], [BBT+18], [LP03], [LV09], [LLHW16]). Unabhängig davon, ob zeitkontinuierliche oder zeitdiskrete Systeme betrachtet werden, wird eine Value Function $V(\boldsymbol{x})$ durch h stetig differenzierbare Basisfunktionen

$\phi(\boldsymbol{x}) \in \mathcal{C}^1$ bzw. eine Q-Function $Q(\boldsymbol{x}, \boldsymbol{u})$ mithilfe h stetig differenzierbarer Basisfunktionen $\phi(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{C}^1$ approximiert. Zusammenfassen der jeweiligen Basisfunktionen ergibt die Vektoren $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^h$ bzw. $\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{u}) \in \mathbb{R}^h$. Am Beispiel der Value Function $V^{\boldsymbol{\mu}}(\boldsymbol{x})$ resultiert bei Verwendung linearer Funktionsapproximatoren[14]

$$V^{\boldsymbol{\mu}}(\boldsymbol{x}) = \boldsymbol{w}^\intercal \boldsymbol{\phi}(\boldsymbol{x}) + \epsilon(\boldsymbol{x}) \tag{2.21}$$

mit dem Gewichtsvektor $\boldsymbol{w} \in \mathbb{R}^h$ und dem Approximationsfehler $\epsilon(\boldsymbol{x})$ (vgl. [VL10], [LVV12], [VL09], [WHL17])[15]. Der Funktionsapproximator einer Value Function oder Q-Function wird in der Literatur als *Critic* bezeichnet, da dieser eine Bewertung eines Regelgesetzes vornimmt (vgl. [BSA83], [LVV12]). Demgegenüber wird ein Funktionsapproximator, der ein Regelgesetz beschreibt, als *Actor* bezeichnet. Zahlreiche ADP-Methoden verwenden lediglich einen Critic (z. B. [WLL14], [LYWW15], [LLW14], [VL09]), beispielsweise, wenn ein analytischer Zusammenhang zwischen der Value Function V oder Q-Function Q und dem (geschätzten) optimalen Regelgesetz genutzt werden kann (vgl. (2.6) und (2.18)). Verfahren, die sowohl V bzw. Q als auch das Regelgesetz $\boldsymbol{\mu}$ durch einen eigenen Funktionsapproximator parametrieren, werden als *Actor-Critic-Methoden* bezeichnet (z. B. [VL10], [BKJ$^+$13], [LW14], [KL15])[16]. Zusammenfassend wird durch Funktionsapproximatoren die Verwendung von wertkontinuierlichen Zustands- und Stellgrößenräumen ermöglicht. Die Suche nach einer optimalen Value- bzw. Q-Function und einem optimalen Regelgesetz wird somit auf eine Parametersuche übertragen.

Schließlich sei noch anzumerken, dass die Formulierung des zu minimierenden Gütemaßes J über einen unendlichen Zeithorizont eine zentrale Bedeutung für die erfolgreiche Verwendung von Funktionsapproximatoren einnimmt, wie die nachfolgende Bemerkung konstatiert.

[14] Wenngleich grundsätzlich auch nichtlineare Funktionsapproximatoren verwendet werden können, so sind im Allgemeinen lineare Funktionsapproximatoren, d. h. eine linear gewichtete Summation ggf. nichtlinearer Basisfunktionen auf theoretischer Ebene besser verstanden und zudem aus praktischer Sicht einfach handhabbar (vgl. [van12], [BT96], [BBT$^+$18], [BBdE10, Kapitel 3.3.1]).

[15] Unter der Annahme, dass die Value Function $V^{\boldsymbol{\mu}}(\boldsymbol{x})$ auf einer kompakten Menge $\mathcal{X}$ stetig differenzierbar ist, lässt sich $\forall x \in \mathcal{X}$ der Approximationsfehler $\epsilon(\boldsymbol{x})$ bei geeignet gewählten Basisfunktionen $\boldsymbol{\phi}(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^h$ mit steigender Anzahl h an Basisfunktionen beliebig verringern. Für polynomielle Basisfunktionen basiert dies beispielsweise auf dem Satz von Weierstraß [Wei85], jedoch existieren auch Verallgemeinerungen (vgl. [HSW90], [BSW97], [AKL05], [VL10]). Da eine sehr große Anzahl an Basisfunktionen und somit zu lernenden Gewichten vermieden werden soll, sei jedoch anzumerken, dass ohne Vorwissen über die Systemdynamik eine praktikable Wahl an Basisfunktionen bei nichtlinearen Systemen grundsätzlich eine bislang ungelöste Herausforderung darstellt (vgl. [BKJ$^+$13, Remark 5]).

[16] In der Literatur wird häufig von einer *Actor-Critic-Struktur* gesprochen, selbst, wenn kein expliziter, zusätzlicher Funktionsapproximator zur Beschreibung der Stellgröße verwendet wird (vgl. [LVV12]). Dies ist darin begründet, dass der Regler $\boldsymbol{\mu}$, der bei allen ADP-Methoden vorhanden ist, in Anlehnung an RL-Methoden als *Actor* bezeichnet werden kann. Zur Abgrenzung wird in der vorliegenden Arbeit der Oberbegriff *verallgemeinerte Critic-Struktur* definiert, um sowohl Methoden mit zusätzlichem, gesondertem Actor-Funktionsapproximator, als auch solche, die direkt aus dem Critic-Gewicht das Regelgesetz $\boldsymbol{\mu}$ bestimmen, zu bezeichnen. Demgegenüber wird explizit dann von *Actor-Critic-Methoden* gesprochen, wenn neben dem Critic-Approximator ein zusätzlicher Funktionsapproximator für das Regelgesetz $\boldsymbol{\mu}$ verwendet wird.

Bemerkung 2.1
Für den Entwurf von ADP-Methoden mithilfe von Funktionsapproximatoren ist das Auftauchen derselben Value Function $V^{\boldsymbol{\mu}}(\cdot)$ *bzw. Q-Function* $Q^{\boldsymbol{\mu}}(\cdot)$ *auf beiden Seiten der zentralen Gleichungen* (2.3a), (2.7) *und* (2.20b) *fundamental. Für diese Eigenschaft ist nach [LVV12, S. 80] maßgeblich die Verwendung eines unendlichen Optimierungshorizontes in* (2.2) *bzw.* (2.11) *verantwortlich.*

2.1.4 ADP-Lösungsansätze

An dieser Stelle lässt sich der Begriff des ADP weiter schärfen. Im Folgenden stellt ADP in Anlehnung an [LWW$^+$17] und [WHL17] eine Kombination aus dynamischer Programmierung, der Verwendung von Funktionsapproximatoren und einer verallgemeinerten Critic-Struktur[17] dar, um mithilfe von Systemtrajektorien Optimalregelungsprobleme zu lösen (vgl. [LV09, S. 39]).

Hierfür werden in diesem Abschnitt grundlegende ADP-Mechanismen eingeführt, die im weiteren Verlauf der vorliegenden Arbeit benötigt werden. Neben der Anpassung des Regelgesetzes $\boldsymbol{\mu}(\boldsymbol{x})$, entweder direkt aus der Value Function V bzw. der Q-Function Q oder über die Anpassung von Actor-Gewichten (vgl. Abschnitt 2.1.3), nimmt insbesondere die Adaption der Critic-Gewichte $\boldsymbol{w}$ eine zentrale Rolle ein. Wird bei Anwendung eines Regelgesetzes $\boldsymbol{\mu}$ die aktuelle Schätzung des Critic-Gewichts angepasst, um die mit diesem Regelgesetz verbundenen Langzeitkosten in Form von $V^{\boldsymbol{\mu}}$ bzw. $Q^{\boldsymbol{\mu}}$ zu beschreiben, so wird dies als *Policy-Evaluation-Schritt* bezeichnet. Eine Anpassung des Regelgesetzes $\boldsymbol{\mu}$ basierend auf der aktuellen Schätzung von $V^{\boldsymbol{\mu}}$ bzw. $Q^{\boldsymbol{\mu}}$, mit dem Ziel, ein bezüglich des Gütefunktionals J verbessertes Regelgesetz zu erhalten, wird hingegen *Policy-Improvement-Schritt* genannt [LVV12]. Je nachdem, ob diese Schritte alternierend oder parallel durchgeführt und ob sie vollständig oder nur teilweise ausgeführt werden, lassen sich ADP-Methoden klassifizieren. Die drei für die vorliegende Arbeit wesentlichen Klassen sind *Policy-Iteration-Algorithmen*, *Value-Iteration-Algorithmen* und *Actor-Critic-Methoden*. Diese werden in den Abschnitten 2.1.4.1–2.1.4.3 thematisiert. Alle drei Klassen sind der sehr allgemeinen Definition der sogenannten *Generalized-Policy-Iteration-*Methoden zuzuordnen, die, unabhängig der Details und Granularität, auf einer irgendwie gearteten Interaktion zwischen Policy-Evaluation- und Policy-Improvement-Prozessen beruhen (vgl. [SB18, Abschnitt 4.6]). Nach der Betrachtung dieser drei zentralen ADP-Klassen wird in Abschnitt 2.1.4.4 schließlich der Unterschied zwischen sogenannten *On-Policy-* und *Off-Policy-ADP-Ansätzen* charakterisiert. Abbildung 2.1 gibt eine Übersicht über die nachfolgend präsentierten ADP-Klassen und deren wichtigste Eigenschaften.

[17] Dies schließt sowohl Actor-Critic-Methoden als auch Ansätze ohne expliziten, zusätzlichen Actor-Funktionsapproximator ein (vgl. Abschnitt 2.1.3). Ansätze, die versuchen, ohne einen Critic-Approximator das Regelgesetz anzupassen (siehe beispielsweise [RPS07], [Mun06]), stehen hingegen nicht im Fokus der vorliegenden Arbeit.

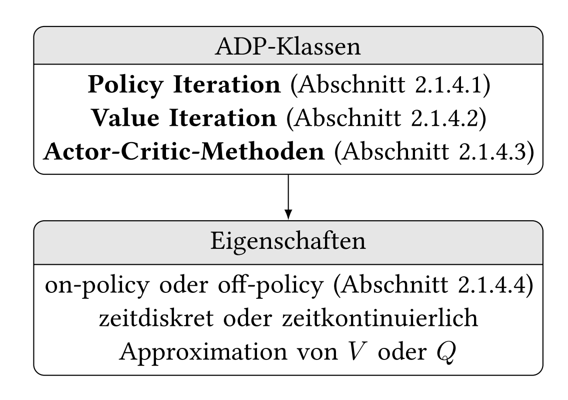


Abbildung 2.1: Klassifikation von ADP-Methoden.

2.1.4.1 Policy Iteration

Policy-Iteration-Algorithmen (PI-Algorithmen) basieren darauf, Regelgesetze auszuwerten, indem deren zugehörige Value Functions bzw. Q-Functions ermittelt werden. Diese Value- bzw. Q-Functions werden dann dazu verwendet, um neue, bezüglich des Gütefunktionals J verbesserte, Regelgesetze zu finden. Dieser Prozess findet iterativ statt [BBdE10, Kapitel 2.4], [SB18, Kapitel 4.3]. PI-Algorithmen sind somit dadurch gekennzeichnet, dass der Policy-Evaluation-Schritt vollständig durchgeführt wird, also stets die Value Function $V^{\boldsymbol{\mu}}$ (bzw. Q-Function $Q^{\boldsymbol{\mu}}$) zu einem Regelgesetz $\boldsymbol{\mu}$ bestimmt wird, bevor das Regelgesetz im Policy-Improvement-Schritt angepasst wird. Am Beispiel einer zeitdiskreten Problemstellung, bei der eine Value Function $V^{\boldsymbol{\mu}}$ approximiert wird, basiert diese iterative Prozedur auf (2.3a) und (2.6). Der prinzipielle Ablauf ist in Algorithmus 2.1 skizziert (vgl. [LVV12], [LV09]). Diese Iteration kann entweder fortgeführt oder durch ein Abbruchkriterium, beispielsweise bei Konvergenz von $V^{[l]}$, beendet werden. Die praktische Auswertung der Schritte 1 und 2 hängt letztlich von der konkreten Problemstellung und den gewählten Lösungsansätzen ab. Für den Policy-Evaluation-Schritt kann beispielsweise ein Gradientenabstieg, ein Batch-Least-Squares- oder ein rekursiver Least-Squares-Ansatz genutzt werden [LVV12]. Zur Lösung des Policy-Improvement-Schrittes kann z. B. direkt die analytische Lösung [LVV12] oder ein Gradientenabstiegsverfahren [LW14] verwendet werden.

Policy-Iteration-Algorithmen existieren neben der in Algorithmus 2.1 gezeigten zeitdiskreten Formulierung, die eine Value Function V lernt (vgl. [LVV12], [LV09], [Hey16]), in zahlreichen Varianten. Beispielsweise kann basierend auf (2.7) und (2.8) für eine zeitdiskrete Systemdarstellung eine Policy-Iteration definiert werden, die eine Q-Function lernt [LP03], [ATLAK07], [BYB94], [LV09]. Erste Ansätze dezentraler und verteilter Q-Learning-Methoden, die ebenfalls auf einer PI basieren, wurden von Görges [Gör19] präsentiert. PI-Algorithmen zur Lösung zeitkontinuierlicher Problemstellungen, die eine Value Function V lernen, müssen entweder im Policy-Evaluation-Schritt die durch (2.13) gegebene Lyapunov-Gleichung lösen [BSW97] oder nutzen die in (2.20b) gegebene IRL-Darstellung [VPAKL09], [LVV12, S. 97]. Analog zu

Algorithmus 2.1 Zeitdiskrete Policy Iteration

1: **Initialisiere** Iterationsindex $l := 0$, zulässiges initiales Regelgesetz $\boldsymbol{\mu}^{[0]}(\boldsymbol{x}_k)$

Schritt 1 *(Policy Evaluation):*

2: Finde $V^{[l+1]}$, sodass gilt:

$$V^{[l+1]}(\boldsymbol{x}_k) = r\left(\boldsymbol{x}_k, \boldsymbol{\mu}^{[l]}(\boldsymbol{x}_k)\right) + \gamma V^{[l+1]}(\boldsymbol{x}_{k+1}). \tag{2.22}$$

Schritt 2 *(Policy Improvement):*

3: Aktualisiere das Regelgesetz

$$\boldsymbol{\mu}^{[l+1]}(\boldsymbol{x}_k) = \underset{\boldsymbol{\mu}(\cdot)}{\arg\min}\left(r(\boldsymbol{x}_k, \boldsymbol{\mu}(\boldsymbol{x}_k)) + \gamma V^{[l+1]}(\boldsymbol{x}_{k+1})\right). \tag{2.23}$$

4: Setze $l := l + 1$. Gehe zu Schritt 1.

IRL-Methoden [VPAKL09] kann eine PI auch so formuliert werden, dass eine Q-Function einer zeitkontinuierlichen Problemstellung iterativ gelernt wird [LPC12].

Die Forderung eines initial zulässigen Regelgesetzes $\boldsymbol{\mu}^{[0]}(\cdot)$ bei PI-Algorithmen kann entweder bei vorhandenem Teilwissen über die Systemdynamik mittels robuster Regelungsansätze erfüllt werden [ML14a, Remark 11], [KLNSK15, Remark 9], [YDZY20] oder vor Anwendung des PI-Algorithmus überprüft werden (vgl. [LW14, Theorem 3.3]). Wenngleich die Forderung nach einem initial zulässigen Regelgesetz eine gewisse Einschränkung[18] von PI-Algorithmen darstellt, so resultiert für nicht-diskontierte Problemstellungen (d. h. für $\gamma = 1$) nach jedem Policy-Improvement-Schritt erneut ein zulässiges Regelgesetz [BSW97, Lemma 9], [LW14]. Auch die Verwendung einer durch einen Funktionsapproximator nicht exakt beschriebenen Value Function V führt in diesem Fall zu einem zulässigen Regelgesetz, wenn die Funktionsapproximation hinreichend genau erfolgt [BSW97, Theorem 26]. Insbesondere ist somit eine exakte Lösung des Policy-Evaluation-Schrittes nicht notwendig, um Stabilität des geschlossenen Regelkreises zu erreichen.

2.1.4.2 Value Iteration

Value-Iteration-Algorithmen (VI-Algorithmen) suchen basierend auf der Bellman-Gleichung oder der HJB-Gleichung iterativ eine optimale Value Function V^* bzw- Q-Function Q^* sowie den optimalen Regler $\boldsymbol{\mu}^*$, führen dabei jedoch keine vollständige Policy Evaluation durch [SB18, Kapitel 4.4], [BBdE10, Kapitel 2.3]. Eine zeitdiskrete Value Iteration, die eine Value Function V lernt, ist in Algorithmus 2.2 gegeben (vgl. [LV09]). Die Schritte *Policy Evaluation*

[18] Hinsichtlich realer Anwendungen erscheint diese Forderung jedoch wenig einschränkend, da ohnehin die Beschränktheit der Systemzustände während der Datenakquise gewährleistet sein muss und ein Betrieb eines Realsystems mit einem instabilen Regelgesetz zu vermeiden ist.

und *Policy Improvement* lassen sich zu einer Gleichung kombinieren (vgl. [LVV12, (30)], [Ber17, (5)]), sodass

$$V^{[l+1]}(\boldsymbol{x}_k) = \min_{\boldsymbol{\mu}(\cdot)} \Big(r\left(\boldsymbol{x}_k, \boldsymbol{\mu}(\boldsymbol{x}_k)\right) + \gamma V^{[l]}(\boldsymbol{x}_{k+1}) \Big) \tag{2.24}$$

resultiert. Im Unterschied zur PI muss das initiale Regelgesetz hierbei nicht zulässig (und damit insbesondere nicht stabilisierend) sein und im Gegensatz zu (2.22) wird auf der rechten Seite von (2.25) die vorherige Schätzung $V^{[l]}(\cdot)$ der Value Function verwendet. Das bedeutet, dass im Policy-Evaluation-Schritt nicht die zum aktuellen Regelgesetz $\boldsymbol{\mu}^{[l]}(\boldsymbol{x}_k)$ gehörende Value Function ermittelt wird, sondern lediglich ein einzelner Update-Schritt in Richtung dieser Value Function[19] vorgenommen wird. Auch hier existieren Formulierungen, die für zeitdiskrete Probleme eine Q-Function [LVV12, S. 95] oder in zeitkontinuierlichen IRL-Ansätzen eine Value Function [LVV12, S. 97] lernen.

VI-Algorithmen weisen den Vorteil auf, prinzipiell nicht mit einem zulässigen Regelgesetz initialisiert werden zu müssen. Zudem erfordert eine einzelne Iteration einer VI, abhängig von der konkreten Implementierung, ggf. einen geringeren Berechnungsaufwand als eine Iteration der PI [KLM96, S. 250][20]. Jedoch kann bei VI-basierten Verfahren, im Gegensatz zu PI-Algorithmen, im Allgemeinen keine Aussage über die Stabilität der Regler $\boldsymbol{\mu}^{[l+1]}$ während der

Algorithmus 2.2 Zeitdiskrete Value Iteration

1: **Initialisiere** Iterationsindex $l := 0$, initiales Regelgesetz $\boldsymbol{\mu}^{[0]}(\boldsymbol{x}_k)$, $V^{[0]} \geq 0$

Schritt 1 *(Policy Evaluation):*

2: Finde $V^{[l+1]}$, sodass gilt:

$$V^{[l+1]}(\boldsymbol{x}_k) = r\left(\boldsymbol{x}_k, \boldsymbol{\mu}^{[l]}(\boldsymbol{x}_k)\right) + \gamma V^{[l]}(\boldsymbol{x}_{k+1}). \tag{2.25}$$

Schritt 2 *(Policy Improvement):*

3: Aktualisiere das Regelgesetz

$$\boldsymbol{\mu}^{[l+1]}(\boldsymbol{x}_k) = \arg\min_{\boldsymbol{\mu}(\cdot)} \Big(r(\boldsymbol{x}_k, \boldsymbol{\mu}(\boldsymbol{x}_k)) + \gamma V^{[l+1]}(\boldsymbol{x}_{k+1}) \Big). \tag{2.26}$$

4: Setze $l := l + 1$. Gehe zu Schritt 1.

[19] Diese Aussage wird insbesondere dadurch gestützt, dass die Bellman-Gleichung eine Fixpunktgleichung ist und die durch $V^{[i+1]}(\boldsymbol{x}_k) = r(\boldsymbol{x}_k, \boldsymbol{\mu}(\boldsymbol{x}_k)) + \gamma V^{[i]}(\boldsymbol{x}_{k+1})$ definierte Iteration mit fixiertem Regelgesetz $\boldsymbol{\mu}(\boldsymbol{x}_k)$ eine Kontraktion darstellt (vgl. [LV09], [Ber20b]). Für ein zulässiges $\boldsymbol{\mu}(\cdot)$ konvergiert diese Iteration für $i \to \infty$ gegen die Lösung des Policy-Evaluation-Schrittes der zeitdiskreten Policy Iteration (2.22) [LV09]. Somit kann (2.25) als einzelner Schritt der für i definierten Iteration mit $\boldsymbol{\mu}(\cdot) = \boldsymbol{\mu}^{[l]}(\cdot)$ interpretiert werden. Dies deckt sich auch mit der Definition der PI und VI für endliche Zustands- und Stellgrößenräume in [SB18, Kapitel 4.3] bzw. [SB18, Kapitel 4.4].

[20] Bei der Anwendung einer PI oder VI auf klassische Markov-Entscheidungsprozesse mit endlichen Zustands- und Aktionsräumen trifft die Aussage zum Komplexitätsunterschied der Berechnung einer einzelnen Iteration z. B. zu (vgl. [KLM96, S. 251]). Bei Verwendung linearer Funktionsapproximatoren und einer blockweisen Least-Squares-Schätzung des Policy-Evaluation-Schrittes kann die Komplexität einer einzelnen Value Iteration ebenfalls reduziert werden, falls die auftretende Pseudoinverse vorab berechnet werden kann (vgl. [BJ16b, Remark 4.1]).

einzelnen Iterationen l getroffen werden, weshalb erst das finale (konvergierte) Regelgesetz für $l \to \infty$ zur Regelung verwendet werden sollte [LW14]. Zudem konvergieren PI-Algorithmen meist in weniger Iterationen l, da die Policy Evaluation eine komplette Bewertung des aktuellen Regelgesetzes vornimmt [LVV12], [Hey16].

2.1.4.3 Actor-Critic-Methoden

Actor-Critic-Methoden (vgl. [KT03], [GBLB12], [SB18, Kapitel 13]) adaptieren die Parameter von Actor- und Critic-Funktionsapproximatoren (vgl. Abschnitt 2.1.3). Diese Actor-Critic-Methoden sind häufig durch PI- oder VI-Algorithmen motiviert und erlauben eine unterschiedliche Granularität der Policy-Evaluation- und Policy-Improvement-Prozesse. Die Adaption des approximierten optimalen Regelgesetzes $\boldsymbol{\mu}$ findet, ähnlich wie bei einer PI oder VI, üblicherweise basierend auf einer geschätzten Value Function oder Q-Function statt. Der Hauptunterschied hierbei ist jedoch, dass bei Actor-Critic-Methoden zumeist gradientenbasierte Methoden verwendet werden, um das Regelgesetz in die durch den Critic bestärkte Richtung anzupassen, anstatt direkt eine vollständige Minimierung basierend auf der aktuellen Schätzung des Critics durchzuführen (vgl. (2.23) und (2.26)).

Beispielsweise präsentieren Vamvoudakis und Lewis [VL10] eine zeitkontinuierliche Actor-Critic-Methode, bei welcher der Critic basierend auf dem quadratischen Fehler der Hamiltonfunktion (vgl. (2.14) mit Verwendung eines Funktionsapproximators) mithilfe eines normierten Gradientenabstiegs und der Actor mit einem modifizierten normierten Gradientenabstieg angepasst wird. Da die Methode von Vamvoudakis und Lewis [VL10] Kenntnis der Eingangsdynamik $\boldsymbol{g}(\boldsymbol{x})$ erfordert, erweitern Bhasin et al. [BKJ$^+$13], die den Critic mithilfe einer zeitkontinuierlichen Least-Squares-Formulierung und den Actor mit einem Gradientenabstieg mit anschließender Projektion adaptieren, die Actor-Critic-Struktur um ein zusätzliches neuronales Netz, das die Systemdynamik identifiziert. Silver et al. [SLH$^+$14] präsentieren eine zeitdiskrete Actor-Critic-Formulierung, bei der die Critic-Parameter einer approximierten Q-Function basierend auf dem Gradienten des quadratischen TD-Fehlers (vgl. (2.7) unter Verwendung eines Funktionsapproximators) adaptiert werden. Die Actor-Gewichte $\boldsymbol{\theta}$, die das Regelgesetz $\boldsymbol{\mu}$ approximieren, werden, beispielsweise mithilfe eines Gradientenabstiegs oder anderer gradientenbasierter Optimierungsverfahren, in Richtung des Gradienten des Gütemaßes J bezüglich des Actor-Gewichtes $\boldsymbol{\theta}$ angepasst. Da dieser Mechanismus später in der Anwendung in Abschnitt 6.1 zum Einsatz kommt, ist eine kurze Einführung in Anhang D.2 gegeben. Darüber hinaus existieren unzählige weitere Actor-Critic-Ansätze. Beispielhaft sei die Arbeit von Li et al. [LGM20] zu nennen, in der ein einzelnes neuronales Netz genutzt wird, um sowohl Gewichte der Value Function[21] als auch Gewichte eines optimiertes Regelgesetzes zu erlernen. Dieser Ansatz kann somit als Verschmelzung der Actor-Critic-Struktur zu einem kombinierten neuronalen Netz interpretiert werden.

[21] Konkret ermöglicht die zusätzliche Nutzung der sogenannten *Advantage Function* (vgl. [WSH$^+$16]), aus der geschätzten Value Function die Q-Function zu schätzen.

2.1.4.4 On-Policy- und Off-Policy-Ansätze

Ein wichtiges Klassifikationsmerkmal von RL- bzw. ADP-Algorithmen ist, ob diese *On-Policy-Algorithmen* oder *Off-Policy-Algorithmen* darstellen. Dabei sind On-Policy-Methoden dadurch charakterisiert, dass die approximierte Value Function V bzw. Q-Function Q die (ggf. diskontierten) Langzeitkosten des während des Adaptionsvorgangs tatsächlich verwendeten Regelgesetzes repräsentieren. Im Gegensatz dazu lernen Off-Policy-Ansätze eine approximierte Value Function V bzw. Q-Function Q, die zu einer sogenannten *Target Policy* gehört. Diese Target Policy weicht im Allgemeinen von der *Behavior Policy* ab, die auf das System angewandt wird, um Messdaten zu generieren (vgl. [KLJ17], [SLH+14], [LCL+19]). Konkret bedeutet das auch, dass das optimale Regelgesetz bei Off-Policy-Ansätzen nicht ausgeführt werden muss, um dieses zu erlernen [van12, S. 17]. Beispiele für On-Policy-Verfahren sind On-Policy-TD-Learning[22] [SB18, Kapitel 6.4] oder die On-Policy-IRL-Methoden von Vrabie et al. [VPAKL09] oder Modares und Lewis [ML14a]. Die Klasse der Off-Policy-Algorithmen schließt beispielsweise klassisches Q-Learning [Wat89], [WD92], Off-Policy-IRL-Algorithmen [JJ12], [JJ14c], [SLW17] den LSPI-Algorithmus [LP03] und Off-Policy-Actor-Critic-Methoden [SLH+14] ein.

Wenn es gelingt, ADP-Methoden off-policy zu entwerfen, bringt dies zwei wesentliche Vorteile mit sich. Zum einen können aufgrund der Off-Policy-Charakteristik Daten beim Training wiederverwendet werden, was die Dateneffizienz signifikant erhöht und essenziell für Anwendungen, beispielsweise in der Robotik, ist [GHLL17]. Diese gezielte Mehrfachverwendung von Datentupeln ist im Bereich des RL unter dem Begriff des *Experience Replay* bekannt, welches maßgeblich für den Erfolg von RL mit tiefen neuronalen Netzen verantwortlich ist [MKS+15], [MKS+13]. Im Gegensatz dazu müssen bei On-Policy-Algorithmen nach jeder Adaption des Regelgesetzes alle bisherigen Trainingsdaten verworfen und neu aufgezeichnet werden. Dies stellt eine datenineffiziente und unpraktikable Einschränkung von On-Policy-Methoden dar [KIP+18]. Zum anderen führen On-Policy-Algorithmen bei der Verwendung von Explorationsrauschen, das zur Systemanregung benötigt wird (vgl. Kapitel 5), zu einem Offset in der Schätzung der Critic-Gewichte und somit auch dem approximierten Regelgesetz [LYD17, Theorem 1], [LCL+19, Lemma 3], [YDZY20, Lemma 1], [KLJ17, Remark 2]. Diesen Offset weisen Off-Policy-Verfahren hingegen nicht auf [LYD17, Theorem 3], [LCL+19, Theorem 3], [YDZY20, Theorem 3], [KLJ17, Theorem 3]. Insbesondere tragen Off-Policy-Ansätze folglich dazu bei, das sogenannte *Exploration-Exploitation-Dilemma* [SB18, S. 3] in den Griff zu bekommen, da die Behavior Policy eine umfassende Exploration ermöglicht, während dennoch eine Adaption in Richtung der optimalen Target Policy stattfinden kann (vgl. [LCL+19], [LHP+16]).

2.2 ADP-basierte Solltrajektorienfolgeregler

Zahlreiche zeitdiskrete (z. B. [LLW+17], [WZL16], [WLLS17], [WHQ20]) und zeitkontinuierliche (z. B. [MCLS02], [KWD16], [Vam17], [Wan20]) ADP-Ansätze betrachten lediglich

[22] Auch unter dem Namen *SARSA* bekannt.

die Regelung des Systemzustands auf eine konstante Ruhelage $\boldsymbol{x} = \boldsymbol{0}$[23] und keine Regelung auf Solltrajektorienverläufe $\boldsymbol{x}_{\mathrm{r},k}$ bzw. $\boldsymbol{x}_{\mathrm{r}}(t)$. Da jedoch die Solltrajektorie die Kosten r und somit auch die Value Function V (vgl. (2.3a) und (2.20)) bzw. Q-Function Q (vgl. (2.7)) beeinflusst, sind diese Regler, die für eine konstante Ruhelage trainiert wurden, nicht für andere Sollzustände geeignet und somit insbesondere nicht auf den Trajektorienfolgeregelungsfall übertragbar.

Des Weiteren weisen Ansätze, die keine explizite funktionale Abhängigkeit der Value Function V oder Q-Function Q von der Referenztrajektorie verwenden, sondern die Form $V^{\boldsymbol{\mu}}(\boldsymbol{x})$ bzw. $Q^{\boldsymbol{\mu}}(\boldsymbol{x}, \boldsymbol{u})$ aufweisen, keine Übertragbarkeit auf eine andere als die während des Trainings verwendete (einzelne, fest vorgegebene) Trajektorie auf. Als Beispiel sei die Arbeit von Yu et al. [YSH$^+$17] zu nennen, bei welcher der Sollzustand kein Argument der Q-Function darstellt, die Kosten r aber dennoch von einem zeitveränderlichen aktuellen Sollzustand abhängen. Yu et al. verwenden in ihrem Actor-Critic-Ansatz während des in zahlreichen Episoden wiederholten Trainingsvorgangs immer wieder dieselbe Solltrajektorie, weshalb diese mittels der Kosten r implizit Einfluss auf die gelernte Q-Function $Q^{\boldsymbol{\mu}}(\boldsymbol{x}, \boldsymbol{u})$ hat. Jedoch trainieren sie ihren Regler damit auf genau diese während des Trainings verwendete Referenztrajektorie, da sowohl $Q(\cdot)$ als auch das gelernte Regelgesetz $\boldsymbol{\mu}$ implizit von dieser Referenz abhängen, aber kein explizites Verständnis über die Abhängigkeit von einer während des Trainingsvorgangs nicht gesehenen Referenztrajektorie aufweisen (vgl. auch (3.4)). Die Verwendung einer anderen Solltrajektorie erfordert somit einen erneuten Trainingsvorgang (vgl. [YSH$^+$17, Abschnitt 5]).

Im Folgenden werden in den Abschnitten 2.2.1–2.2.3 unterschiedliche Klassen ADP-basierter Solltrajektorienfolgeregelungsansätze aus der Literatur vorgestellt und in Abschnitt 2.2.4 eine kurze Übersicht der Anwendung von ADP-Methoden auf reale regelungstechnische Anwendungen gegeben, um die erste zentrale Forschungslücke herauszuarbeiten.

2.2.1 ADP-Ansätze unter Nutzung der dynamischen Inversion

Zahlreiche ADP-basierte Methoden nutzen einen auf einer Zustandslinearisierung (vgl. beispielsweise [Kha02, Kapitel 13]) basierenden Vorsteuerungsentwurf, der sich in der Literatur unter dem Begriff einer *dynamischen Inversion* findet [EBHS07]. Diese Konzepte bestimmen üblicherweise a priori und unter vollständiger Kenntnis oder nach vorheriger Identifikation der Systemdynamik einen inversionsbasierten Vorsteuerterm und nutzen erst dann ADP-Verfahren, um lediglich den Zustandsrückführungsterm des Reglers zu adaptieren. Somit wird genau genommen kein datenbasierter ADP-Solltrajektorienfolgeregler gelernt, da der Vorsteuerterm modellbasiert berechnet wird. Tabelle 2.1 gibt eine Übersicht über diese Methoden. Die Spalte *globale Diskontierung* γ gibt hierbei Aufschluss darüber, ob ein Diskontierungsfaktor γ

[23] Wenngleich diese Methoden prinzipiell auch für einen konstanten Sollzustand ungleich null trainiert werden könnten, so ist dieser Sollzustand zur Laufzeit nicht veränderbar und auch kein expliziter Parameter des gelernten Regelgesetzes $\boldsymbol{\mu}$. Vielmehr wird während des Trainings implizit der Einfluss des einmalig fest gewählten Sollzustands auf V bzw. Q gelernt. Dies kann letztlich als Festwertregelung (vgl. [Lun20a, S. 362]) mit der zusätzlichen wesentlichen Einschränkung, dass die Führungsgröße nicht verändert werden kann, d. h. ohne äußere Eingriffsmöglichkeit, interpretiert werden.

im Gütemaß verwendet wird (vgl. (2.2)), um die Value- bzw. Q-Function endlich zu halten, oder welche anderen Mechanismen hierfür genutzt werden.

Zhang et al. [ZWL08] stellen einen zeitdiskreten Ansatz vor, der eine Value Iteration nutzt. Huang und Liu [HL14] präsentieren ebenfalls eine zeitdiskrete Methode. Neben einem neuronalen Netz zur Approximation der Systemdynamik verwenden sie einen Actor-Critic-Ansatz und passen sowohl die Critic-Gewichte als auch den Funktionsapproximator zur Beschreibung des Zustandsrückführungsterms mithilfe eines Gradientenabstiegs an. Auch die zeitkontinuierliche ADP-Trajektorienfolgeregelungsmethode von Kamalapurkar et al. [KDBD15] verwendet eine derartige dynamische Inversion sowie ein Actor-Critic-Verfahren.

Ein Nachteil all dieser Methoden ist jedoch die im Allgemeinen nicht erfüllte Annahme, dass die Eingangsmatrix $\boldsymbol{g}(\boldsymbol{x})$ bzw. $\boldsymbol{B}$ invertierbar[24] sein muss. Zudem muss die gesamte Systemdynamik entweder bekannt sein oder identifiziert werden (vgl. [KLM+14], [KLL15]). Mu et al. [MSWS17] fordern bei ihrem Ansatz für zeitkontinuierliche, eingangsaffine Systeme ebenfalls, dass die durch $\boldsymbol{f}(\boldsymbol{x})$ und $\boldsymbol{g}(\boldsymbol{x})$ beschriebene Systemdynamik bekannt ist, ihr Ansatz erlaubt jedoch immerhin eine beschränkte und zustandsabhängige Unsicherheit in der Systemdynamik. Dierks und Jagannathan [DJ09] nutzen in ihrem gradientenbasierten Actor-Critic-Ansatz

Veröffentlichung	Systemmodell erforderlich (oder identifiziert)	zeitkontinuierlich (k) oder zeitdiskret (d)	Systemdynamik	Value Function V oder Q-Function Q	Grundalgorithmus	globale Diskontierung γ	dynamische Inversion
[ZWL08]	ja	d	eingangsaffin	V	VI	nein[c]	ja
[HL14]	ja	d	eingangsaffin	V	AC[a]	nein[c]	ja
[KDBD15]	ja	k	eingangsaffin	V	AC[b]	nein[c]	ja
[MSWS17]	ja[d]	k	eingangsaffin	V	PI	nein[c]	ja
[DJ09]	teilw.	d	eingangsaffin	V	AC[a]	nein[c]	ja

[a] Actor-Critic mit (ggf. normiertem) Gradientenabstieg.
[b] Actor-Critic, Critic mit zeitkontinuierlichem Least-Squares-Update mit Vergessensfaktor.
[c] Gütemaß bestraft Abweichung der Stellgröße vom Vorsteuerungsanteil.
[d] Jedoch Unsicherheit im Modell erlaubt.

Tabelle 2.1: Übersicht über ADP-Trajektorienfolgeregelungsmethoden, die auf dem Konzept der dynamischen Inversion basieren.

[24] Im Sinne der Existenz einer Matrix $\tilde{\boldsymbol{g}}(\boldsymbol{x})$, sodass $\boldsymbol{g}(\boldsymbol{x})\tilde{\boldsymbol{g}}(\boldsymbol{x}) = \boldsymbol{I}$ gilt (vgl. beispielsweise [DJ09], [DJ10]).

neben Funktionsapproximatoren zur Schätzung des Critics und des Zustandsrückführungsterms noch einen dritten Funktionsapproximator[25], um auch den Vorsteuerterm zu adaptieren. Dennoch muss auch bei diesem Verfahren $\boldsymbol{g}(\boldsymbol{x})$ bekannt und invertierbar sein. Abschließend sei noch angemerkt, dass bei all diesen Konzepten, die den Ansatz der Zustandslinearisierung nutzen, im Gütefunktional $J(\cdot)$ die quadratische Abweichung der Stellgröße vom Vorsteuerterm und somit nur die transiente Stellgröße und nicht die eigentliche Stellenergie bestraft wird. Ein geschlossener optimierungsbasierter Ansatz liegt somit nicht vor.

2.2.2 Globale Solltrajektorienvorgabe durch eine Exosystemdynamik ohne externe Eingriffsmöglichkeit

Eine andere in der ADP-Literatur weitverbreitete Annahme ist, dass die Referenztrajektorie global einer (häufig unbekannten) Exosystemdynamik folgt [ML13], [ML14a], [ML14b], [ZZXS17], [KL15], [LLHW16], [KLM$^+$14], [LYD17], [Vam16], [KLNSK15], [QZL13], [GJ16], [GJ15], [BA18], [KLL15]. Basierend auf dieser Annahme wird der Systemzustand um einen Exosystemzustand erweitert, sodass auf bestehende ADP-Mechanismen zurückgegriffen werden kann. Eine Übersicht über ADP-Trajektorienfolgeregelungsmethoden, die eine solche globale Vorgabe der Referenztrajektorie annehmen, ist in Tabelle 2.2 gegeben. Diese Ansätze werden im Folgenden diskutiert.

Modares und Lewis [ML13] stellen für zeitkontinuierliche linear-quadratische optimale Trajektorienfolgeregelungsprobleme einen Policy-Iteration-Algorithmus vor, der jedoch die vollständige Kenntnis der erweiterten Systemdynamik benötigt. Darüber hinaus existieren sowohl zeitkontinuierliche [ML14a], [ML14b], [ZZXS17] als auch zeitdiskrete [KL15] Ansätze, welche die Value Function V approximieren und zwar keine Kenntnis über die interne Systemdynamik $\boldsymbol{f}(\boldsymbol{x})$ bzw. $\boldsymbol{A}$ benötigen, die Eingangmatrix $\boldsymbol{g}(\boldsymbol{x})$ bzw. $\boldsymbol{B}$ jedoch bekannt sein muss. Diese Einschränkung einer bekannten Eingangsmatrix weisen die Ansätze von Luo et al. [LLHW16] (zeitdiskrete, nichtlineare Systemdynamik), Kiumarsi et al. [KLM$^+$14] und Li et al. [LYD17] (jeweils zeitdiskrete, lineare Systemdynamik), sowie Vamvoudakis [Vam16] (zeitkontinuierliche, lineare Systemdynamik), die jeweils eine Q-Function lernen, nicht auf. Kiumarsi et al. [KLNSK15] betrachten eine lineare, zeitdiskrete Systemdynamik und lernen eine ADP-basierte optimale Ausgangsrückführung, indem der Systemzustand aus einer endlichen Anzahl vergangener Werte der Stellgrößen, Ausgangsgrößen und Sollzustände rekonstruiert wird und anschließend alternativ entweder eine Policy Iteration oder eine Value Iteration verwendet wird. Qin et al. [QZL13] stellen für zeitkontinuierliche linear-quadratische Ausgangsfolgeregelungsprobleme eine IRL-basierte Policy Iteration vor, die keine Kenntnis der Werte der Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ benötigt. Hierbei wird jedoch einschränkend angenommen, dass die Anzahl der Exosystemzustände identisch zur Anzahl der Ausgangsgrößen des zu regelnden Systems ist. Gao und Jiang [GJ16], [GJ15] verwenden eine auf dem Prinzip des IRL basierende Policy Iteration, um linear-quadratische Trajektorienfolgeregelungsprobleme mit zeitkontinuierlicher Systemdynamik und unbekannten Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ zu lösen. Bernhard und Adamy [BA18] betrachten Trajektorienfolgeregelungsprobleme mit linearer,

[25] Dieser schätzt letztlich implizit die interne Dynamik $\boldsymbol{f}(\cdot)$.

Veröffentlichung	Systemmodell erforderlich (oder identifiziert)	zeitkontinuierlich (k) oder zeitdiskret (d)	Systemdynamik	Value Function V oder Q-Function Q	Grundalgorithmus	globale Diskontierung γ	dynamische Inversion
[ML13]	ja	k	linear	V	PI	ja	nein
[ML14a]	teilw.	k	linear	V	PI	ja	nein
[ML14b]	teilw.	d	eingangsaffin	V	AC[a]	ja	nein
[ZZXS17]	teilw.	k	eingangsaffin	V	PI	ja	nein
[KL15]	teilw.	d	eingangsaffin	V	AC[a]	ja	nein
[LLHW16]	nein	d	nichtlinear	Q	PI	ja	nein
[KLM+14]	nein	d	linear	Q	PI	ja	nein
[LYD17]	nein	d	linear	Q	PI	ja	nein
[Vam16]	nein	k	linear	Q	AC[a]	nein[b]	nein
[KLNSK15]	nein	d	linear	V	PI/VI	ja	nein
[QZL13]	nein	k	linear	V	PI	nein[b]	nein
[GJ16], [GJ15]	nein	k	linear	V	PI	nein[c]	nein
[BA18]	nein	k	linear	V	PI	nein[d]	nein
[KLL15]	teilw.	d	eingangsaffin	V	PI/VI	ja	nein

[a] Actor-Critic mit (ggf. normiertem) Gradientenabstieg.
[b] Exosystem muss stabil sein.
[c] Gütemaß bestraft Abweichung der Stellgröße vom Vorsteuerungsanteil.
[d] Unendliche Kosten erlaubt.

Tabelle 2.2: Übersicht über ADP-Trajektorienfolgeregelungsmethoden, die eine globale Vorgabe der Referenztrajektorie annehmen.

unbekannter, zeitkontinuierlicher Systemdynamik und unendlichen Kosten[26]. Kiumarsi et al. [KLL15] behandeln zeitdiskrete, eingangsaffine, zeitveränderliche Systeme. Hierbei werden für unterschiedliche Systemdynamiken unterschiedliche Value Functions gelernt, anschließend wird mittels sogenannter *adaptive self-organizing maps* automatisiert zwischen diesen Repräsentationen gewechselt.

[26] Hierbei wird die Definition der überholenden Optimalität (vgl. [Ber20a, Definition 2.3], (engl.): *overtaking optimality*) verwendet.

Alle im vorliegenden Abschnitt diskutierten Arbeiten weisen jedoch einen entscheidenden Nachteil auf: die Annahme, dass der Exosystemzustand ausgehend von dessen Initialzustand propagiert wird und zur Laufzeit nicht durch eine externe Eingriffsmöglichkeit verändert werden kann[27]. Die Referenztrajektorie ist daher global vorgegeben und nicht von außen beeinflussbar[28]. Daher können beispielsweise Nutzereingaben, Straßenverläufe oder gewünschte Systemzustandsverläufe, die aus einer übergeordneten Ebene resultieren, nicht berücksichtigt werden. Als illustratives Beispiel, um die damit verbundene Problematik zu veranschaulichen, werde ein Fahrzeug betrachtet, das einem beliebigen, realen Straßenverlauf folgen soll. Um mit diesem Fahrzeug beispielsweise von Karlsruhe nach Kaiserslautern zu fahren, müsste unter Verwendung der in diesem Abschnitt diskutierten Methoden der gesamte Straßenverlauf auf dieser Strecke durch die Exosystemdynamik und dessen Anfangszustand beschrieben sein. Während dies bereits eine nicht handhabbare Annahme darstellt, müsste zudem für jede andere Fahrtstrecke ein geeignetes Exosystem gefunden und der Regler erneut trainiert werden. Damit sind mit den bestehenden Methoden insbesondere auch keine beliebigen Solltrajektorien als Eingabe der Regelungsmethoden möglich. Diese Eigenschaft ist in Abbildung 2.2 veranschaulicht und stellt erhebliche Einschränkungen für die allgemeine Anwendbarkeit dieser adaptiven optimalen Trajektorienfolgeregelungen dar.

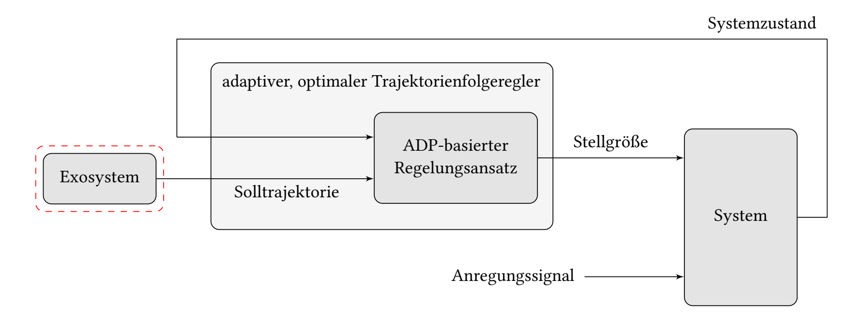


Abbildung 2.2: Schematische Darstellung von Literaturansätzen, die annehmen, die Referenztrajektorie folge global einer Exosystemdynamik. Eine flexible Beeinflussbarkeit des Solltrajektorienverlaufs von außen fehlt hierbei.

2.2.3 Stationäre Sollzustandsvorgabe

Eine Alternative zur Annahme, die Referenztrajektorie werde global durch eine Exosystemdynamik erzeugt, stellt die Erweiterung des Systemzustands um den aktuellen Sollzustand oder die Abweichung vom aktuellen Sollzustand [HSSH17], [PRH19], [PKRH20], [WXL$^+$14], bzw. die Erweiterung um einen projizierten aktuellen Sollzustand [NKJS04] dar. Shi et al. [SSW18] führen diese Idee fort, indem der Systemzustand nicht nur um den Sollzustand im

[27] Andernfalls würde die Markov-Eigenschaft verletzt und die Trainingsdaten wären nicht mit der Bellman-Gleichung (2.5) bzw. HJB-Gleichung (2.16) verträglich (vgl. [van12], Abschnitt 3.1 und Abschnitt 4.1).

[28] Dies kann als Servoregelung (vgl. [Lun20a, S. 398], [AM89, S. 89]) ohne äußere Eingriffsmöglichkeit interpretiert werden, bei der das Exosystem die Führungsgröße generiert.

aktuellen Zeitschritt, sondern auch um den Sollzustand des direkt nachfolgenden Zeitschritts ergänzt wird. Zwar lassen die ADP-basierten Methoden, welche den aktuellen (und ggf. nachfolgenden) Sollzustand in der Value Function bzw. Q-Function explizit berücksichtigen, die Vorgabe beliebiger stationärer Sollzustände zu (vgl. Abbildung 2.3), jedoch wird der zeitliche Verlauf der Solltrajektorie hierbei nicht (oder im Fall von Shi et al. [SSW18] nur sehr eingeschränkt) berücksichtigt. Vielmehr repräsentiert die Value Function $V^{\boldsymbol{\mu}}(\boldsymbol{x}_k, \boldsymbol{x}_{\mathrm{r},k})$ nur die unter dem Regelgesetz $\boldsymbol{\mu}$ entstehenden Langzeitkosten, wenn sich das System im Zustand $\boldsymbol{x}_k$ befindet und auf den stationären Endwert $\boldsymbol{x}_{\mathrm{r},k}$ eingeregelt werden soll, was jedoch nicht dem nach (2.11) zu minimierenden Gütefunktional entspricht. Der Regler, der eine Festwertregelung (vgl. [Lun20a, S. 362]) darstellt, reagiert damit lediglich auf Abweichungen zum aktuellen Sollzustand, berücksichtigt jedoch nicht den zukünftigen Verlauf der Referenztrajektorie. Dies kann zu einem zeitlichen Versatz zwischen dem Istzustand und dem Sollzustand führen.

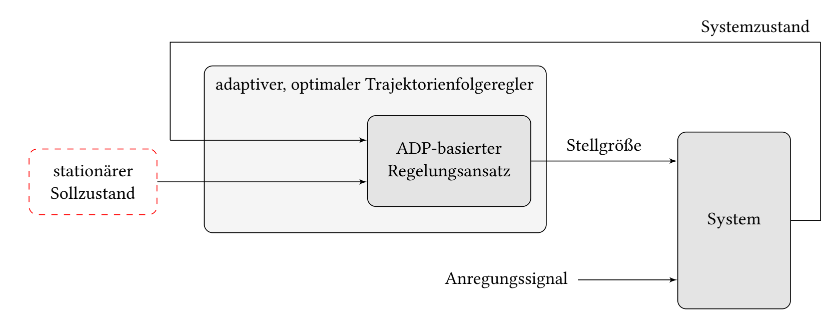


Abbildung 2.3: Schematische Darstellung ADP-basierter Optimalregelungskonzepte aus der Literatur, welche die Vorgabe eines stationären Sollzustands verwenden.

2.2.4 ADP in realen regelungstechnischen Anwendungen

Nach diesen Einblicken in unterschiedliche Klassen ADP-basierter Solltrajektorienfolgeregler wird im Folgenden ein kurzer Überblick hinsichtlich bisheriger realer ADP-Anwendungen gegeben. Obwohl zahlreiche Publikationen zu ADP-basiertem Reglerentwurf und zu ADP-basierten Zustandsfolgereglern existieren, betrachten die meisten dieser Veröffentlichungen lediglich Simulationsergebnisse, jedoch keine reale Anwendung der Methoden im Experiment. Insofern ist die empirische Basis dünn und vernachlässigt praktische Herausforderungen, z. B. aufgrund von Mess- und Stellgliedungenauigkeiten. Als prominente Beispiele seien [LV09], [WLMZ18], [JJ14c], [BKJ+13], [ML14a], [LLHW16] und [KLM+14] genannt. Gleiches gilt für das in Abschnitt 6.1 betrachtete Anwendungsbeispiel ADP- oder RL-basierter Geschwindigkeitsregler für Fahrzeuge. Auch hier existieren zumeist nur Simulationsergebnisse (siehe [PRH19], [BK18], [NCH08], [DCd11] und [KBJ+21, Table II]) oder Laborergebnisse anhand eines Fahrsimulators [PT12]. Huang et al. [HXH+19] arbeiten zwar mit einem Realfahrzeug, trainieren ihren Geschwindigkeitsregler jedoch basierend auf einem stark vereinfachten, datenbasierten Modell der Longitudinaldynamik. Aus diesem Modell werden Stichproben

gezogen, anhand derer ein Längsregler trainiert wird. Zudem findet keine Berücksichtigung des zukünftigen Solltrajektorienverlaufs statt, der Adaptionsprozess erfolgt nicht anhand von Realdaten während der Fahrt und dynamische Vorgänge werden ignoriert, d. h. interne, nicht messbare Systemgrößen werden nicht rekonstruiert (vgl. [PRH19]).

Auch die Betrachtung anderer Anwendungsbeispiele liefert ein ähnliches Bild. Li et al. [LGM20] demonstrieren beispielsweise mithilfe von Simulationsergebnissen, wie ADP genutzt werden kann, um das Energiemanagement in hybridelektrischen Fahrzeugen zu optimieren. Li und Görges [LG19], [LG20] stellen Ansätze für modellfreie, ADP-basierte Abstandsregeltempomaten vor und präsentieren Simulationsergebnisse. Shi et al. [SSW18] untersuchen einen RL-basierten Regler für ein Unterwassergefährt, wobei eine stationäre Sollzustandsvorgabe (vgl. Abschnitt 2.2.3) verwendet wird. Die dortigen Ergebnisse beschränken sich auf Simulationen, eine reale Anwendung findet nicht statt. Hwangbo et al. [HSSH17] stellen einen RL-basierten Regler vor, der neuronale Netze trainiert und die Position eines Quadrocopters regelt. Dem Regler wird dabei lediglich eine Sollposition vorgegeben (Festwertregelung), an welcher sich der Quadrocopter stabilisieren soll. Eine Berücksichtigung des Verlaufs der Solltrajektorie findet nicht statt. Auch die von Ng et al. [NKJS04] präsentierte RL-basierte Regelung eines Modellhelikopters verwendet als Vorgabe nur die aktuelle Abweichung von einer Wunschposition, anstatt den Sollverlauf zu übergeben. Sowohl Hwangbo et al. [HSSH17] als auch Ng et al. [NKJS04] nutzen zwar ihre durch einen RL-basierten Ansatz trainierten Regler nach dem Trainingsvorgang an realen Systemen, jedoch basiert das Erlernen des Regelgesetzes jeweils auf Reinforcement Learning anhand von Daten, die aus einem Simulationsmodell erzeugt wurden und nicht direkt mittels Messdaten. Somit muss ein geeignetes Systemmodell für das Training vorliegen, also zunächst modelliert oder identifiziert werden. Es bleibt ungeklärt, inwiefern eine Adaption des Reglers direkt basierend auf realen Messdaten möglich ist. Die in Kapitel 1 erwähnten Arbeiten [KIP+18], [ABC+20] erfordern eine enorme Menge an Real- bzw. Simulationsdaten für einen erfolgreichen Trainingsvorgang. Zudem wird in [KIP+18] im Fall eines erfolgreichen Greifversuchs des Roboters eine binäre Belohnung ausgeschüttet und in [ABC+20] nur die Abweichung von einer stationären Soll-Orientierung eines in einer Roboterhand befindlichen Würfels bestraft und kein Solltrajektorienverlauf explizit vorgegeben.

Insbesondere eine reale Anwendung eines modellfreien ADP-Solltrajektorienreglers, der nicht nur den aktuellen Sollzustand, sondern einen flexiblen Sollzustandsverlauf berücksichtigt, und dabei strukturelles Vorwissen über das zugrunde liegende Problem ausnutzt, ist somit bis zum jetzigen Zeitpunkt in der Literatur nicht vorhanden. Aufgrund von Unterschieden zwischen Simulationen und Realanwendungen (vgl. [DAMH19], [CHL19], [Bro92]) ist aus einer anwendungsorientierten regelungstechnischen Perspektive eine Validierung der direkten Anwendung der ADP-Methoden auf reale Systeme jedoch unverzichtbar.

2.2.5 Fazit

Bislang existierende ADP-Solltrajektorienfolgeregelungsansätze lassen sind in drei verschiedene Gruppen klassifizieren. Inversionsbasierte Methoden nach Abschnitt 2.2.1 erfordern

Kenntnis über die Systemdynamik und bestrafen nicht die eigentliche Stellenergie. Daher werden sie nachfolgend nicht weiter für den ADP-Kontext betrachtet. Ansätze, die annehmen, dass die Referenztrajektorie einer globalen Exosystemdynamik folgt (vgl. Abschnitt 2.2.2), weisen bislang keine äußere Beeinflussungsmöglichkeit dieses Solltrajektorienverlaufs auf (vgl. Abbildung 2.2). Die Vorgabe eines stationären Endwertes nach Abschnitt 2.2.3 erlaubt zwar eine Sollwertvorgabe von außen (vgl. Abbildung 2.3), der weitere zeitliche Verlauf der Referenztrajektorie wird hierbei jedoch nicht berücksichtigt. Abschnitt 2.2.4 unterstreicht zudem, dass bislang nur wenige reale Anwendungen ADP-basierter Regelungsansätze betrachtet wurden und diese insbesondere keine flexible Solltrajektorienvorgabe erlauben.

Somit kann konstatiert werden, dass ein generalisierender Ansatz (vgl. Abbildung 2.4), der aus der externen Vorgabe einer Solltrajektorie eine für die Verwendung durch ADP-Ansätze kompatible[29] Approximation generiert, die zugleich Informationen über den zeitlichen Verlauf der Referenztrajektorie beinhaltet, bis dato nicht existiert.

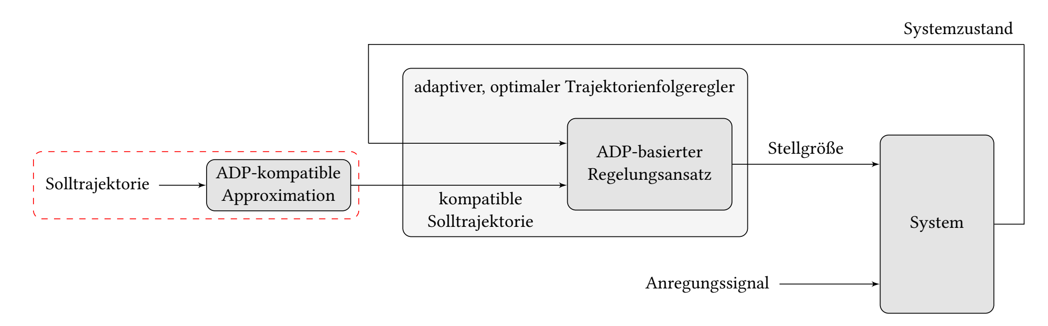


Abbildung 2.4: Schematische Darstellung eines generalisierenden Ansatzes, der die externe Vorgabe eines Solltrajektorienverlaufs erlaubt und diese Referenztrajektorie auf eine mit dem ADP-Formalismus kompatible Weise approximiert. Eine solche Methode, die vielfältige, flexible Solltrajektorien in den ADP-Mechanismus integrieren kann, existiert bislang nicht.

2.3 Anregung ADP-basierter Regelungsansätze

Für Konvergenzaussagen von Systemidentifikationsmethoden sowie adaptiven und ADP-basierten Algorithmen wird stets die Annahme gefordert, dass das betrachtete System ausreichend angeregt ist. Am Beispiel des ADP lässt sich anschaulich plausibilisieren, weshalb eine geeignete Anregung des Systems erforderlich ist. So lassen sich beispielsweise bei einem System, das sich dauerhaft in einer stabilen Ruhelage $\boldsymbol{x} = \boldsymbol{0}$ befindet, keine umfassenden Informationen über dessen dynamisches Verhalten gewinnen. Sämtliche Regelgesetze mit $\boldsymbol{\mu}(\boldsymbol{0}) = \boldsymbol{0}$ erscheinen dann aus Sicht eines zu minimierenden Gütefunktionals gleichermaßen

[29] Eine ADP-kompatible Solltrajektoriendarstellung erlaubt die Integration eines Sollzustandsverlaufs in ADP-basierte Regelungsansätze. Eine genaue Definition erfolgt durch Definition 3.1 für den zeitdiskreten Fall und durch Definition 4.1 für den zeitkontinuierlichen Fall.

optimal, da der Fall $\boldsymbol{x} \neq \mathbf{0}$ nicht beurteilt werden kann. Erst durch die Betrachtung unterschiedlicher Zustands- und Stellgrößenkombinationen, d. h. durch eine geeignete *Exploration* (vgl. [SB18, S. 3]) können Informationen darüber gewonnen werden, welche Regelgesetze hinsichtlich des Gütefunktionals zu bevorzugen sind.

Eine Formalisierung derartiger Anregungsannahmen wird unter dem Begriff der sogenannten *Persistent-Excitation-Bedingung*[30] zusammengefasst. Dieser Begriff entstand in den 1960er-Jahren zunächst im Kontext der Systemidentifikation [ÅB66] (vgl. [NA05, S. 238], [BU16]), spielt jedoch auch bei adaptiven Regelungsansätzen [NA05, S. 239], [IS96] und ADP-Ansätzen [VLV13, S. 62] eine zentrale Rolle bei der Betrachtung von Konvergenzeigenschaften.

Konkret wird für zeitkontinuierliche Signale $\boldsymbol{\sigma}(t) \in \mathbb{R}^h$ die Einhaltung einer PE-Bedingung

$$\int_t^{t+T} \boldsymbol{\sigma}(\tau)\boldsymbol{\sigma}^\mathsf{T}(\tau)\,\mathrm{d}\tau \succeq \alpha \boldsymbol{I}, \tag{2.27}$$

$\alpha, T \in \mathbb{R}_{>0}$, $\forall t \geq t_0$, gefordert (vgl. [NA05, Definition 6.2], [PSA17, Definition 4]). Falls das Signal $\boldsymbol{\sigma}(t)$ diese PE-Bedingung erfüllt, so wird es im Folgenden als *PE-Signal* bezeichnet. Die Relevanz der PE-Bedingung (2.27) offenbart sich anhand der häufig auftretenden Differenzialgleichung

$$\frac{\mathrm{d}\widetilde{\boldsymbol{w}}(t)}{\mathrm{d}t} = -\boldsymbol{\sigma}(t)\boldsymbol{\sigma}^\mathsf{T}(t)\widetilde{\boldsymbol{w}}(t), \tag{2.28}$$

wobei $\widetilde{\boldsymbol{w}}(t) \in \mathbb{R}^h$ den Fehler eines zu schätzenden Parameters $\boldsymbol{w} \in \mathbb{R}^h$ darstellt. Für die gleichmäßige asymptotische Stabilität der Ruhelage von (2.28) ist es notwendig und hinreichend, dass $\boldsymbol{\sigma}(t)$ ein PE-Signal ist [NA05, S. 246]. Anschaulich interpretiert genügt aufgrund von $\mathrm{Rang}(\boldsymbol{\sigma}(t)\boldsymbol{\sigma}^\mathsf{T}(t)) \leq 1$ die in $\boldsymbol{\sigma}(t)$ zu einem einzelnen Zeitpunkt t enthaltene Information nicht, um die h Unbekannten des Parametervektors $\boldsymbol{w}$ zu bestimmen. Die PE-Bedingung (2.27) fordert daher, dass das Integral der positiv semidefiniten Matrix $\boldsymbol{\sigma}(\tau)\boldsymbol{\sigma}^\mathsf{T}(\tau)$ über ein endliches Intervall $[t, t+T]$, $\forall t \geq t_0$, positiv definit ist (vgl. [NA05, S. 247]). Eine zweite anschauliche Interpretation ergibt sich wie folgt: Nach [NA05, S. 247] ist das Integral des Betrags der Projektion eines PE-Signals $\boldsymbol{\sigma}(t)$ entlang jedes beliebigen Einheitsvektors des $\mathbb{R}^h$ über ein endliches Intervall $[t, t+T]$, $\forall t \geq t_0$, ungleich null. Dies bedeutet, dass in diesem endlichen Intervall das Signal $\boldsymbol{\sigma}(t)$ eine Basis des $\mathbb{R}^h$ durchlaufen muss (vgl. [Kar19, S. 42]).

Die Erfüllung der PE-Bedingung für den Regressor $\boldsymbol{\sigma}(t)$ ist für Konvergenzaussagen bei der zeitkontinuierlichen Parameteradaption [Ngu18, S. 132], [JALG18], [Pra17], der adaptiven Regelung [NA05, Abschnitt 6.5.2], [BS86], [AG08], [LK98], [NMH$^+$15] und bei ADP-Ansätzen [VL10], [VL11], [Vam17], [JKBD15], [ML14b], [VVL09b], [TCTH19], [BKJ$^+$13], [LLW14], [LYWW15], [ZCZL11], [SWL19, Assumption 6.4], [ZZXS17], [ZDJ14] zentral.

Das zeitdiskrete Pendant der durch (2.27) formulierten PE-Bedingung ist durch

$$\sum_{i=k}^{k+T_\mathrm{d}} \boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^\mathsf{T} \succeq \alpha \boldsymbol{I} \succ \mathbf{0} \tag{2.29}$$

[30] In seltenen Fällen ist auch die deutschsprachige Bezeichnung *fortwährende Erregung* [BU16] zu finden.

mit $T_\mathrm{d} \in \mathbb{N}$, $T_\mathrm{d} > 0$, gegeben [Bit84], [Mar84], [GM86]. Diese Bedingung findet sich unter anderem bei rekursiven zeitdiskreten ADP-Algorithmen [ATLAK07], [LLHW16], [BYB94], [ZQJL14], aber auch bei zeitkontinuierlichen ADP-Ansätzen, die mithilfe von IRL-Methoden gelöst werden [BJ16a], [FC16].

Zu erwähnen seien an dieser Stelle außerdem Arbeiten, die zur PE-Bedingung äquivalente [PSA17] und alternative [CJ10], [KKD14], [KWD13] Anregungsbedingungen vorstellen. Die unter dem Begriff des *concurrent learning* aufzufindenden Methoden der adaptiven Regelung [CJ10] bzw. ADP-Literatur [KKD14], [KWD13] fordern hierbei die Erfüllung einer Rangbedingung an zuvor abgespeicherte Daten. Zwar sind diese Rangbedingungen ggf. in der Anwendung nachträglich komfortabel überprüfbar, jedoch bleibt auch bei diesen Ansätzen ungeklärt, wie Systeme angeregt werden sollten, um geeignete Daten zu erzeugen. Somit wird auch hier das Vorhandensein geeignet angeregter Daten a priori vorausgesetzt (vgl. [CJ10, Condition 1], [KKD14, Assumption 1]).

Da in Kapitel 5 eine Betrachtung der zeitkontinuierlichen PE-Bedingung nach (2.27) stattfindet, konzentriert sich die nachfolgende Übersicht auf diesen Fall. Für die Erfüllung der PE-Bedingung spielt dabei insbesondere der Frequenzgehalt von $\boldsymbol{\sigma}$ eine wesentliche Rolle (vgl. [SB89, Abschnitt 4.3], [LK98], [NA87]). So ist beispielsweise ein skalares Signal $\sigma_\mathrm{s} \in \mathbb{R}$ genau dann ein PE-Signal, wenn dessen Leistungsdichtespektrum mindestens eine Frequenzlinie aufweist [NA05, S. 253], [Kar19, Lemma 4.2]. Für den Fall $\boldsymbol{\sigma} \in \mathbb{R}^h$ sind Aussagen für den Zusammenhang zwischen dem Frequenzspektrum von $\boldsymbol{\sigma}$ und der Erfüllung der PE-Bedingung in [BS83, Lemma 3.4] gegeben. Bei der Analyse von Signalen ist zudem die Frage, unter welchen mathematischen Operationen die PE-Eigenschaft eines Signals erhalten bleibt, relevant. Die bestehende Literatur zu Systemidentifikationsmethoden und adaptiver Regelung offenbart hierbei, dass für *lineare* algebraische und dynamische Transformationen von Signalen theoretische Aussagen hinsichtlich der PE-Eigenschaft existieren (vgl. [NA05, S. 249 ff.], [BS86]). Ein Beispiel für eine solche algebraische Signaltransformation ist, dass bei linearen Abbildungen eines Signals durch Matrizen $\boldsymbol{M} \in \mathbb{R}^{m_1 \times m_2}$ mit Maximalrang und $m_1 \leq m_2$ dessen PE-Eigenschaft erhalten bleibt [NA05, Lemma 6.1]. Als Beispiel für eine dynamische Transformation lässt sich nennen, dass der Systemzustand eines linearen, Eingangs-Ausgangs-stabilen, steuerbaren dynamischen Systems n-ter Ordnung mit skalarer Eingangsgröße genau dann PE ist, wenn das Leistungsdichtespektrum des skalaren Eingangssignals mindestens n Frequenzlinien aufweist [BS86], [NA05, S. 255 ff.]. Aussagen für mehrdimensionale Eingangsgrößen sind in [MB90] und [GM86] zu finden.

Bei ADP-basierten Ansätzen treten jedoch selbst bei scheinbar einfachen Problemstellungen Nichtlinearitäten in den Basisfunktionen $\boldsymbol{\phi}(\boldsymbol{x}(t))$ der linearen Funktionsapproximatoren (vgl. Abschnitt 2.1.3) auf. Ein Beispiel ist durch LQ-Optimierungsprobleme gegeben (vgl. [PLB15, S. 303]). Da diese Nichtlinearitäten den Frequenzgehalt der betrachteten Signale verändern können, und sich aus dem Frequenzgehalt des Systemzustands $\boldsymbol{x}(t)$ somit keine direkten Rückschlüsse auf den Frequenzgehalt der Basisfunktion $\boldsymbol{\phi}(\boldsymbol{x}(t))$ (vgl. (2.21)) bzw. deren zeitliche Ableitungen $\frac{\mathrm{d}\boldsymbol{\phi}(\boldsymbol{x}(t))}{\mathrm{d}t}$ [31] ziehen lassen, sind existierende theoretische Erkenntnisse nicht ohne

[31] Gerade für das Signal $\frac{\mathrm{d}\boldsymbol{\phi}(\boldsymbol{x}(t))}{\mathrm{d}t}$ ist jedoch die Erfüllung der PE-Bedingung von entscheidender Bedeutung für die Konvergenz des Critic-Gewichts $\boldsymbol{w}$ (vgl. [VL10, Theorem 1]).

Weiteres auf den ADP-Kontext übertragbar. Abhängig von den konkreten Nichtlinearitäten können beispielweise zusätzliche Frequenzen auftreten und sich positiv auf die Erfüllung der PE-Eigenschaft auswirken oder Frequenzen können sich gegenseitig aufheben und somit die PE-Bedingung verletzen (vgl. [LK98], [LK99]). Eine sorgfältige Analyse des Einflusses der vorhandenen Nichtlinearitäten auf die PE-Eigenschaft ist daher insbesondere im ADP-Kontext unumgänglich.

Unter den wenigen in der Literatur verfügbaren Ansätzen, welche die PE-Bedingung im Zusammenhang mit Nichtlinearitäten analysieren, sind die Arbeiten von Lin und Kanellakopoulos [LK98] (System in strenger Ausgangs-Rückkopplungsform), [LK99] (System in strenger parametrischer Rückkoplungsform) sowie von Adetola und Guay [AG06] (zustandslinearisierbare Systeme mit linearer Parameterabhängigkeit), [AG08] (nichtlineare Systeme mit linearer Parameterabhängigkeit) zu nennen, die jedoch dem Bereich der adaptiven Regelung zuzuordnen sind. Allgemeine Untersuchungen und theoretische Analysen hinsichtlich einer geeigneten Anregung im ADP-Kontext existieren bislang nicht [JZLH19], [JKBD15], [KLL15], [BKJ+13].

Obwohl eine geeignete Anregung insbesondere auch bei ADP-Methoden stets erforderlich ist, wird in vorhandenen Publikationen üblicherweise die Erfüllung einer auf die jeweils konkrete Formulierung zugeschnittenen PE-Bedingung als gegeben vorausgesetzt. Eine methodische Analyse findet bislang nicht statt. In der Hoffnung, die zur Konvergenz der jeweiligen Methode erforderliche PE-Bedingung zu erfüllen, sind in der Literatur unterschiedliche heuristische Strategien zu finden. So wird häufig Explorationsrauschen in Form von Zufallssignalen auf die Stellgröße addiert, wie beispielsweise Gaußsches weißes Rauschen oder, seltener, auf einem Intervall gleichverteiltes Rauschen. Häufig ist dieses Anregungssignal jedoch nicht näher spezifiziert und die Erfüllung der Anregungsbedingung wird nicht weiter untersucht. Eine ähnlich verbreitete Methode ist die Addition eines Anregungssignals, das aus einer Summation und Multiplikation von Sinus- und Kosinusfunktionen resultiert[32], auf die Stellgröße. Eine Untersuchung, wie diese Sinus- und Kosinussignale zu wählen sind, sodass eine geeignete Anregung erreicht wird, findet jedoch bislang nicht statt. Schließlich schlagen manche Autorinnen und Autoren als dritte Strategie vor, das System für verschiedene Zustände auszuwerten, beispielsweise durch eine wiederholte Neuinitialisierung des Anfangszustands. Auch hier findet keine theoretische Untersuchung statt und es bleibt ungeklärt, wie eine solche Neuinitialisierung konkret erfolgen müsste, um die Erfüllung der PE-Bedingung zu gewährleisten. Beispiele für ADP-Ansätze, die diese drei Heuristiken zur Anregung verwenden, sind Tabelle 2.3 zu entnehmen.

Bei der Betrachtung der Anregungssignale für ADP-Methoden ist schließlich noch hervorzuheben, dass mögliche anwendungsspezifische Anforderungen bislang, insbesondere bei Verwendung weißen Rauschens und anderer Zufallssignale zur Anregung, nicht berücksichtigt werden. Beispielsweise weisen reale physikalische und biologische Systeme üblicherweise Tiefpassverhalten auf [EP10], hochfrequente Stellsignale werden also gedämpft und sind somit

[32] Da sich Produkte von Sinus- und Kosinusfunktionen stets durch Summen aus Sinus- und Kosinusfunktionen darstellen lassen, wird im Folgenden nur noch von Summationen gesprochen.

additives Zufallssignal auf Stellgröße:
[ATLAK07], [BK18], [DJ09], [DJ10], [DJ11], [FY16], [JKBD15], [KL15], [KLM+14], [KLNSK15], [LCL+19], [LLW14], [LP03], [LPC12], [LVV12], [QZLY19], [Vam17], [VMKL17], [WLMZ18], [WLZZ16], [WY18], [ZCL13], [ZDJ15], [ZQJL14], [ZZWZ16]
additiver Summenterm aus Sinus- und Kosinusfunktionen auf Stellgröße:
[BA18], [BKJ+13], [FC16], [GJ16], [JJ14b], [JJ14c], [JJ12], [JKBD15], [KDBD15], [KLJ17], [KLL15], [LYD17], [LYWW15], [LNY+15], [MRLP16], [ML14b], [QZLY19], [TCTH19], [VMKL17], [VMH16], [Vam16], [Vam15], [WLL14], [YHL13], [YLLL16]
wiederholte Neuinitialisierung des Systemzustands:
[DLL+19], [LLW14], [VL09], [VPAKL09], [WL12]

Tabelle 2.3: Anregungsheuristiken verschiedener ADP-Ansätze.

potenziell ungeeignet zur Anregung. Zudem ist zu beachten, dass die Verwendung ungeeigneten Explorationsrauschens eine erhöhte mechanische Belastung technischer Systeme mit sich bringt und zur schnelleren Abnutzung oder sogar Beschädigung des Systems führen kann (vgl. [KBP13], [dKTB18]). So beansprucht die Verwendung weißen Rauschens durch ruckartige Stelleingriffe potenziell die Aktuatorik. Schließlich ist eine solch ruckartige Anregung für Adaptionsverfahren, die sich online zur Laufzeit anpassen sollen, meist unerwünscht und beispielsweise auch in einer möglichen zukünftigen Anwendung im Mensch-Maschine-Kontext weder komfortabel noch zur Anregung unbekannter Reaktionen des Menschen geeignet[33].

Zusammenfassend kann konstatiert werden, dass zahlreiche Publikationen zu ADP-basierten Verfahren die Erfüllung geeigneter Anregungsbedingungen voraussetzen, bis dato jedoch noch keine verifizierbare Methode existiert, welche die Einhaltung der PE-Bedingung im nichtlinearen Kontext sicherstellt (vgl. [JZLH19], [JKBD15], [KLL15], [BKJ+13]). Dies ist insbesondere auf den Mangel an allgemeinen theoretischen Aussagen zur PE-Bedingung unter nichtlinearen Signaltransformationen, die im Bereich des ADP aufgrund nichtlinearer Gütefunktionale selbst bei LQ-Optimierungsproblemen auftreten, zurückzuführen. Im Hinblick auf eine spätere Übertragbarkeit auf reale Anwendungsbeispiele sind zudem problemspezifische

[33] Im Kontext der Mensch-Maschine-Interaktion, bei der sich Mensch und Maschine gegenseitig bei einer Regelungsaufgabe unterstützen, kann der Mensch aufbauend auf Arbeiten zur Modellierung des sensomotorischen Systems [Sco04], [Tod04], [JJ14a] als Optimalregler modelliert werden. Im Zusammenspiel mit einer ebenfalls optimierungstheoretisch formulierten Automation ergibt sich ein Differenzialspiel (vgl. auch Definition 5.1) [FFH17], [NC15]. Da üblicherweise sowohl das Regelgesetz als auch das Gütefunktional des menschlichen Handlungspartners unbekannt ist, muss entweder das Regelgesetz (vgl. beispielsweise [KNFH20]) oder das Gütefunktional identifiziert werden (vgl. [KIR+17], [IKFH17], [IBKH20], [RIK+17]), oder die Automation adaptiert sich mithilfe eines ADP-basierten Ansatzes aus der Interaktion mit dem Menschen. Da jedoch das menschliche neuromuskuläre System Tiefpasscharakteristik aufweist [KAV+14], sind hochdynamische Anregungssignale hierbei ungeeignet, um adäquate Informationen zur Adaption der Automation zu erhalten. Eine sinnvoll gewählte Anregung ist also insbesondere auch für zukünftige adaptive Mensch-Maschine-Konzepte erforderlich.

Anforderungen zu beachten, wie beispielsweise Freiheitsgrade beim Anregungsdesign zur Berücksichtigung von Tiefpasseigenschaften.

2.4 Wissenschaftliche Fragestellungen und Beiträge der Arbeit

Ausgehend von den Forschungslücken, die in den vorigen Abschnitten aufgezeigt wurden, werden im Folgenden zwei konkrete Fragestellungen abgeleitet, zu denen die vorliegende Arbeit Beiträge liefert. Außerdem wird eine Übersicht über diese wissenschaftlichen Beiträge gegeben.

2.4.1 ADP-kompatible, flexible Solltrajektoriendarstellung

ADP-basierte Regelungsansätze sind üblicherweise auf der Bellman-Gleichung bzw. Hamilton-Jacobi-Bellman-Gleichung begründet und für einen unendlichen Optimierungshorizont formuliert. Eine wesentliche Herausforderung beim Entwurf von ADP-Solltrajektorienfolgereglern ist eine mit der Bellman- bzw. HJB-Gleichung kompatible Repräsentation der Solltrajektorie. Bislang existieren aber weder eine allgemeine theoretische Definition ADP-kompatibler Solltrajektoriendarstellungen noch geeignete Ansätze ADP-basierter Solltrajektorienfolgeregler, welche die Einbeziehung einer von außen vorgebbaren Darstellung variabler Solltrajektorienverläufe ermöglichen. Daraus ergibt sich die folgende Fragestellung:

Forschungsfrage 1:
Wie können Solltrajektorienverläufe dargestellt werden, um in einen modellfreien ADP-Formalismus integrierbar zu sein, der einen approximierten optimalen Solltrajektorienfolgeregler mithilfe von Messdaten eines dynamischen Systems und ohne Verwendung eines Systemmodells erlernt?

Beitrag 1:
In dieser Arbeit wird erstmalig der Begriff der ADP-kompatiblen Solltrajektoriendarstellung eingeführt und formal definiert. Weiterhin werden neuartige und mit dem ADP-Formalismus kompatible Darstellungsformen flexibler, von außen vorgebbarer Referenztrajektorien für Sollzustände präsentiert sowie deren Eigenschaften analysiert. Insbesondere beinhaltet die vorliegende Arbeit:

- Theoretische Beiträge in zeitdiskreter Darstellung (Kapitel 3):
  - Eine neuartige, mit dem ADP-Formalismus kompatible, flexible, parametrierte Darstellungsform der Referenztrajektorie, die insbesondere die Beschreibung des Solltrajektorienverlaufs in einer lokalen Umgebung des aktuellen Zeitpunkts erlaubt. Des Weiteren werden Existenz und Eindeutigkeit des optimalen Regelgesetzes für den LQ-Solltrajektorienfolgeregelungsfall analysiert. Zudem wird ein neues Stabilitätskriterium präsentiert. Schließlich wird ein Vorgehen vorgestellt, das es ermöglicht, ADP-basierte Regler, die diese parametrierte Referenztrajektorie nutzen, anhand von Messdaten zu trainieren.
  - Die im ADP-Kontext erstmalige direkte Verwendung der Sollzustände auf einem endlichen, gleitenden Vorausschauhorizont. Hierbei wird für den LQ-Fall die Existenz und Eindeutigkeit sowie die exakte Form der optimalen Lösung analysiert. Letzteres ist essenziell, um effiziente Funktionsapproximatoren mit einer möglichst geringen Anzahl zu schätzender Gewichte definieren und mithilfe von Messdaten trainieren zu können. Außerdem wird die Konvergenz der verwendeten ADP-Methode für den Solltrajektorienfolgeregelungsfall bewiesen.
- Theoretische Beiträge in zeitkontinuierlicher Darstellung (Kapitel 4):
  - Eine neuartige, mit dem ADP-Formalismus kompatible, flexible, parametrierte Repräsentation der Referenztrajektorie, die sich aus der Superposition der Lösungen gewöhnlicher Differenzialgleichungen ergibt. Existenz, Eindeutigkeit und Stabilität der optimalen Lösung werden analysiert. Schließlich wird ein Konzept vorgestellt, das diese Solltrajektoriendarstellung in eine ADP-Methode integriert.
- Reale Anwendungsbeispiele eines zuvor entwickelten ADP-basierten Solltrajektorienfolgereglers (Kapitel 6):
  - Ein Geschwindigkeitsregler für ein Realfahrzeug, dessen Abhängigkeit von einer parametrierten Beschreibung des Sollgeschwindigkeitsverlaufs direkt aus Messdaten erlernt wird. Insbesondere findet die Adaption hierbei online statt, d. h. die Reglergewichte werden während der Fahrt im geschlossenen Regelkreis angepasst.
  - Ein ADP-basierter Solltrajektorienfolgeregler für ein reales Ball-auf-Platte-System. Dieser wird einerseits mit einem ADP-basierten Sollzustandsregler (Festwertregelung) und andererseits mit einem modellbasierten optimalen Reglerentwurf verglichen.

2.4.2 Konvergenz ADP-basierter Regelungsansätze

Bei sämtlichen adaptiven und ADP-basierten Regelungsverfahren wird eine ausreichende Anregung des Systems bzw. der für die Adaption relevanten Signale gefordert, um Konvergenz der zu schätzenden Parameter gegen die jeweiligen Zielgrößen zu gewährleisten. Die Signale, auf denen diese Adaption im ADP-Kontext beruht und für welche die Erfüllung der

PE-Eigenschaft gefordert wird, resultieren aus nichtlinearen Transformationen der Systemzustände (vgl. Abschnitt 2.3), woraus die folgende Forschungsfrage resultiert:

Forschungsfrage 2:
Welche Bedingungen an die dynamischen Zustände eines Systems gewährleisten, dass die PE-Eigenschaft eingehalten wird und ADP-Methoden konvergieren?

Beitrag 2:
In Kapitel 5 dieser Arbeit wird eine weitverbreitete PE-Eigenschaft für zeitkontinuierliche ADP-Ansätze (vgl. (2.27)) analysiert. Hierbei werden eingangsaffine Nicht-Nullsummen-Differenzialspiele, die eine Generalisierung eingangsaffiner Optimalregelungsprobleme darstellen, betrachtet. Die Einhaltung der PE-Eigenschaft garantiert die Konvergenz der Critic-Gewichte bei Adaption mit einem Gradientenabstiegsverfahren. Wie Abschnitt 2.3 motiviert, spielen die in einem Signal vorhandenen Frequenzen eine wesentliche Rolle bei der Erfüllung der PE-Eigenschaft. Daher wird in der vorliegenden Arbeit untersucht, wie der Frequenzgehalt des Systemzustands durch die Verwendung polynomieller Basisfunktionen zur Critic-Funktionsapproximation beeinflusst wird. Der zentrale Beitrag ist schließlich durch neuartige hinreichende Frequenzbedingungen an den Systemzustand gegeben, welche gewährleisten, dass die für die Konvergenz benötigte PE-Eigenschaft erfüllt ist. Die präsentierten Bedingungen weisen zudem Freiheitsgrade bezüglich der für die Anregung verwendeten Frequenzen und Amplituden auf. Somit können anwendungsspezifische Anforderungen berücksichtigt werden. Ausgehend von den theoretischen Erkenntnissen werden Simulationsergebnisse präsentiert und diskutiert.

3 Zeitdiskrete ADP-basierte Solltrajektorienfolgeregelung

Dieses Kapitel beantwortet die Forschungsfrage, wie Solltrajektorienverläufe dargestellt werden können, um in zeitdiskrete, modellfreie ADP-Ansätze integrierbar zu sein (vgl. Forschungsfrage 1). Hierzu wird in Abschnitt 3.1 zunächst definiert, wodurch eine ADP-kompatible Solltrajektoriendarstellung charakterisiert ist. Anschließend werden zwei Ansätze präsentiert: In Abschnitt 3.2 wird eine zeitdiskrete, parametrierte ADP-kompatible Solltrajektoriendarstellung vorgestellt, anschließend betrachtet Abschnitt 3.3 die Verwendung beliebiger Sollzustände auf einem endlichen Vorausschauhorizont. Ein abschließendes Resümee rundet das Kapitel ab.

3.1 Definition ADP-kompatibler zeitdiskreter Trajektorien

In diesem Abschnitt werden allgemeine Vorüberlegungen zur Funktionsweise von ADP-Methoden vorgenommen, um darauf aufbauend Anforderungen an den Entwurf ADP-basierter Trajektorienfolgeregler abzuleiten. Hierzu wird zunächst exemplarisch die zeitdiskrete Value Function

$$V^{\boldsymbol{\mu}}(\boldsymbol{x}_k) = \sum_{\kappa=k}^{\infty} \gamma^{\kappa-k} r(\boldsymbol{x}_\kappa, \boldsymbol{\mu}(\boldsymbol{x}_\kappa)) \tag{3.1a}$$

$$= r(\boldsymbol{x}_k, \boldsymbol{\mu}(\boldsymbol{x}_k)) + \gamma V^{\boldsymbol{\mu}}(\boldsymbol{x}_{k+1}) \tag{3.1b}$$

$$= r(\boldsymbol{x}_k, \boldsymbol{\mu}(\boldsymbol{x}_k)) + \gamma V^{\boldsymbol{\mu}}(\boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{g}(\boldsymbol{x}_k)\boldsymbol{\mu}(\boldsymbol{x}_k)) \tag{3.1c}$$

(vgl. (2.3a)) ohne Verwendung einer Solltrajektorie betrachtet. Jedoch lassen sich die hier gewonnenen Erkenntnisse auch auf Formulierungen unter Verwendung einer Q-Function (vgl. (2.7)) oder auf zeitkontinuierliche Darstellungen (vgl. (2.13) und (2.20b)) übertragen. Eine zentrale Eigenschaft der Value Function $V^{\boldsymbol{\mu}}(\boldsymbol{x}_k)$ ist, dass eine funktionale Abhängigkeit vom aktuellen Zustand $\boldsymbol{x}_k$ die (ggf. diskontierten) Gesamtkosten repräsentieren kann, die über einen unendlichen Zeithorizont anfallen. Die Betrachtung der Value Function in (3.1) offenbart, dass dies maßgeblich der Markov-Eigenschaft des Systemzustands $\boldsymbol{x}_k$ zu verdanken ist, der aus der Dynamik

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{g}(\boldsymbol{x}_k)\boldsymbol{\mu}(\boldsymbol{x}_k) \tag{3.2}$$

des geschlossenen Regelkreises resultiert (vgl. [van12], [SLH$^+$14]). Diese Repräsentation der Gesamtkosten in Form der Value Function $V^{\boldsymbol{\mu}}(\boldsymbol{x}_k)$ ist mit dem betrachteten System (2.1),

dem mit der Value Function assoziierten Regelgesetz $\boldsymbol{\mu}(\boldsymbol{x}_k)$ und der Einschrittkostenfunktion $r(\boldsymbol{x}_k, \boldsymbol{u}_k)$ verknüpft. Ohne diese Markov-Eigenschaft würde der Systemzustand $\boldsymbol{x}_k$ nicht als alleiniger, expliziter Parameter der Value Function $V^{\boldsymbol{\mu}}(\boldsymbol{x}_k)$ genügen, um die Kosten über einen unendlichen Zeithorizont korrekt abbilden zu können, da aus dem aktuellen Zustand $\boldsymbol{x}_k$ der weitere Zustandsverlauf, der implizit in $V^{\boldsymbol{\mu}}(\boldsymbol{x}_k)$ enthalten ist, nicht bestimmt werden könnte[34].

Aufgrund dieser Markov-Eigenschaft ist es basierend auf (3.1b) zudem grundsätzlich möglich, mithilfe von Datentupeln

$$\mathcal{T}_k := \{\boldsymbol{x}_k, \boldsymbol{\mu}(\boldsymbol{x}_k), r(\boldsymbol{x}_k, \boldsymbol{\mu}(\boldsymbol{x}_k)), \boldsymbol{x}_{k+1}\} \tag{3.3}$$

(und ggf. γ) die Value Function $V^{\boldsymbol{\mu}}(\boldsymbol{x}_k)$, bzw. bei Verwendung eines Funktionsapproximators (vgl. (2.21)) das Critic-Gewicht $\boldsymbol{w}$, zu bestimmen[35] (vgl. beispielsweise den Policy-Evaluation-Schritt der PI (2.22) oder der VI (2.25)).

Um im Folgenden Solltrajektorienverläufe $\boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{x}_{\mathrm{r},k+1}, \dots$ in das Gütefunktional J und somit die Value Function $V^{\boldsymbol{\mu}}$ zu integrieren, hängen die Einschrittkosten r neben dem Systemzustand $\boldsymbol{x}_k$ und der Stellgröße $\boldsymbol{u}_k$ im Allgemeinen zusätzlich vom Sollzustand $\boldsymbol{x}_{\mathrm{r},k}$ ab und es ergibt sich $r(\boldsymbol{x}_k, \boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{u}_k)$. Würde eine Abhängigkeit von der Solltrajektorie $\boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{x}_{\mathrm{r},k+1}, \dots$ nicht explizit in der Value Function verwendet und weiterhin eine Funktion der Form $V^{\boldsymbol{\mu}}(\boldsymbol{x}_k)$ zugrunde gelegt, wie beispielsweise in der Arbeit von Yu et al. [YSH+17] (vgl. Abschnitt 2.2), so müsste diese Value Function $\forall \boldsymbol{x}_k$ die Gleichung

$$V^{\boldsymbol{\mu}}(\boldsymbol{x}_k) = \sum_{\kappa=k}^{\infty} \gamma^{\kappa-k} r(\boldsymbol{x}_\kappa, \boldsymbol{x}_{\mathrm{r},\kappa}, \boldsymbol{\mu}(\boldsymbol{x}_\kappa)) = r(\boldsymbol{x}_k, \boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{\mu}(\boldsymbol{x}_k)) + \gamma V^{\boldsymbol{\mu}}(\boldsymbol{x}_{k+1}) \tag{3.4}$$

erfüllen (vgl. (3.1)). Dies wäre jedoch bei beliebigen Sollzuständen $\boldsymbol{x}_{\mathrm{r},k}$ nicht möglich, da eine Abhängigkeit der Sollzustände nicht von der Value Function $V^{\boldsymbol{\mu}}(\boldsymbol{x}_k)$ erfasst werden könnte. Für den sehr eingeschränkten Spezialfall, dass $\boldsymbol{x}_{\mathrm{r},k}$ stets eindeutig durch eine zeitinvariante Abbildung $\boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{x}} : \mathbb{R}^n \to \mathbb{R}^n$ durch $\boldsymbol{x}_{\mathrm{r},k} = \boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{x}}(\boldsymbol{x}_k)$ mit dem Systemzustand $\boldsymbol{x}_k$ verknüpft ist, ließe sich eine Value Function $V^{\boldsymbol{\mu}}(\boldsymbol{x}_k)$ definieren, die (3.4) erfüllt. Die damit gelernte Value Function wäre jedoch implizit nur für diesen direkt mit $\boldsymbol{x}_k$ korrelierten Sollzustandsverlauf gültig und in der Anwendung kaum von Nutzen.

Den Sollzustand nur für den aktuellen und $n_\mathrm{h} < \infty$ weitere Zeitschritte vorzugeben, wäre hingegen mit einem Optimierungsproblem mit endlichem Zeithorizont verknüpft, da der

---

[34] Ebenso ist es für klassische Methoden des RL, die das Lösen Markovscher Entscheidungsprozesse [Bel57b] anstreben, erforderlich, dass der Systemzustand $\boldsymbol{x}_k$ die Markov-Eigenschaft erfüllt [SB18, S. 49], [KT03]. Bei Betrachtung stochastischer Markov-Entscheidungsprozesse ist somit die Zustandsübergangswahrscheinlichkeit nur vom aktuellen, nicht jedoch von vergangenen Zuständen abhängig.

[35] Da es sich bei (3.1b) um eine skalare Gleichung handelt, sind zur Schätzung eines h-dimensionalen Critic-Gewichts mindestens h unabhängige Datentupel erforderlich (vgl. beispielsweise [LVS12, S. 495]). Insbesondere kann der Critic unter Einbeziehung genügend vieler und entsprechend vielseitiger Datentupel $\mathcal{T}_k$ (d. h. bei ausreichender Anregung des Systems, vgl. Abschnitt 2.3) aus den während des Trainingsprozesses betrachteten Zustandsübergängen $\boldsymbol{x}_k, \boldsymbol{\mu}(\boldsymbol{x}_k), \boldsymbol{x}_{k+1}$ implizit die Systemdynamik und über die Einschrittkosten $r(\boldsymbol{x}_k, \boldsymbol{\mu}(\boldsymbol{x}_k))$ die zugrunde liegenden Gesamtkosten erlernen.

weitere Verlauf der Solltrajektorie $\boldsymbol{x}_{\mathrm{r},\kappa}$ für $\kappa > k + n_{\mathrm{h}}$ unbekannt wäre[36]. Die Verwendung eines unendlichen Zeithorizontes ist jedoch nach Bemerkung 2.1 dafür verantwortlich, dass in (3.1) auf beiden Seiten der Gleichung dieselbe Value Function $V^{\boldsymbol{\mu}}(\cdot)$ auftaucht und diese somit durch denselben Funktionsapproximator beschrieben werden kann[37].

Würde eine Value Function eines Optimierungsproblems mit unendlichem Optimierungshorizont gesucht, die einen komplett beliebigen Solltrajektorienverlauf berücksichtigt, so müsste diese eine Abhängigkeit beliebiger Sollzustände $\boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{x}_{\mathrm{r},k+1}, \ldots$ bis ins Unendliche aufweisen, also durch

$$\begin{aligned} V^{\boldsymbol{\mu}}(\boldsymbol{x}_k, \boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{x}_{\mathrm{r},k+1}, \ldots) &= \sum_{\kappa=k}^{\infty} \gamma^{\kappa-k} r\big(\boldsymbol{x}_\kappa, \boldsymbol{x}_{\mathrm{r},\kappa}, \boldsymbol{\mu}(\boldsymbol{x}_\kappa, \boldsymbol{x}_{\mathrm{r},\kappa}, \boldsymbol{x}_{\mathrm{r},\kappa+1}, \ldots)\big) \\ &= r\big(\boldsymbol{x}_k, \boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{\mu}(\boldsymbol{x}_k, \boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{x}_{\mathrm{r},k+1}, \ldots)\big) \\ &\quad + \gamma V^{\boldsymbol{\mu}}(\boldsymbol{x}_{k+1}, \boldsymbol{x}_{\mathrm{r},k+1}, \boldsymbol{x}_{\mathrm{r},k+2}, \ldots) \end{aligned} \tag{3.5}$$

definiert sein (vgl. [KPRH20]). Eine Value Function dieser Form wäre jedoch weder aus theoretischer noch aus praktischer Sicht handhabbar: Weder lassen sich unendlich viele Eingabeparameter verwenden, noch ist in den meisten praktischen Anwendungen der Sollverlauf (wie beispielsweise die Sollgeschwindigkeit oder Sollposition eines Fahrzeugs) bis ins Unendliche bekannt. Auch würde eine Value Function mit unendlich vielen beliebigen Referenzparametern mit einem unendlichdimensionalen Funktionsapproximator (vgl. Abschnitt 2.1.3) korrelieren.

Somit offenbart sich, dass der Verlauf der Solltrajektorie zwar als explizite Abhängigkeit in die Value Function aufgenommen werden muss, dies aber mithilfe einer endlichdimensionalen Repräsentation erfolgen sollte, um handhabbar zu sein. Sei diese endlichdimensionale Repräsentation im Zeitschritt k allgemein durch einen n_ζ-dimensionalen ($n_\zeta < \infty$) Parameter $\boldsymbol{\zeta}_k \in \mathcal{Z} \subset \mathbb{R}^{n_\zeta}$ mit Definitionsbereich $\mathcal{Z}$ und $\boldsymbol{x}_{\mathrm{r},k} = \boldsymbol{f}_{\boldsymbol{x}_{\mathrm{r}},\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k)$ gegeben, wobei $\boldsymbol{f}_{\boldsymbol{x}_{\mathrm{r}},\boldsymbol{\zeta}} : \mathcal{Z} \to \mathcal{X}$ eine zeitinvariante Abbildung darstellt. Damit nun $\boldsymbol{\zeta}_k$ genügt, um den zeitlichen Verlauf von $\boldsymbol{x}_{\mathrm{r},\kappa}, \forall \kappa \geq k$, erfassen zu können, muss zudem für $\boldsymbol{\zeta}_k$ die Markov-Eigenschaft gelten, also der Parameter $\boldsymbol{\zeta}_{k+1} = \boldsymbol{f}_{\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k)$ mit $\boldsymbol{f}_{\boldsymbol{\zeta}} : \mathcal{Z} \to \mathcal{Z}$ zum Folgezeitpunkt lediglich vom aktuellen Parameter $\boldsymbol{\zeta}_k$ abhängen[38]. Aus praktischer Sicht lässt sich durch den Parameter $\boldsymbol{\zeta}_k$ somit in jedem Zeitschritt k Information über den aktuellen Solltrajektorienverlauf im ADP-Mechanismus berücksichtigen. Beim Entwurf von $\boldsymbol{f}_{\boldsymbol{x}_{\mathrm{r}},\boldsymbol{\zeta}}(\cdot)$ und $\boldsymbol{f}_{\boldsymbol{\zeta}}(\cdot)$ kann, abhängig von der Anwendung und den Anforderungen, nicht zuletzt durch die Wahl der Dimension n_ζ des Parameters $\boldsymbol{\zeta}$ ein Kompromiss zwischen der Komplexität und der Kompaktheit der

36 Wird $\forall \kappa > k + n_{\mathrm{h}}$ ein konstanter Sollzustand angenommen, so lässt sich hingegen wieder ein Optimierungsproblem mit unendlichem Optimierungshorizont formulieren. Dies wird in Abschnitt 3.3 vorgestellt.

37 Die Verwendung desselben Funktionsapproximators auf beiden Seiten der Bellman-Gleichung würde bei einem Optimierungsproblem mit endlichem Optimierungshorizont zu einem systematischen Fehler führen. Bei sehr langem aber endlichem Optimierungshorizont oder bei einer starken Diskontierung ($\gamma \ll 1$) würde dieser systematische Fehler zwar reduziert, jedoch führt ersteres zu einem hochdimensionalen und zudem schwierig trainierbaren Funktionsapproximator und letzteres verfälscht einerseits ggf. das eigentliche Optimierungsziel und kann andererseits ein instabiles Gesamtsystem begünstigen (vgl. Abschnitt 4.2.2).

38 Bzw. in zeitkontinuierlichen Fall die Änderungsrate $\dot{\boldsymbol{\zeta}}(t)$ lediglich von $\boldsymbol{\zeta}(t)$ abhängen (vgl. Abschnitt 4.1).

Solltrajektoriendarstellung eingegangen werden. Zusammenfassend lässt sich eine mit dem ADP-Formalismus kompatible Solltrajektoriendarstellung wie folgt definieren:

Definition 3.1 (Zeitdiskrete ADP-kompatible Solltrajektoriendarstellung)
Eine mit dem ADP-Formalismus kompatible Solltrajektoriendarstellung mit dem Sollzustand $\boldsymbol{x}_{\mathrm{r},k} \in \mathcal{X}$ *ist durch zeitinvariante Funktionen* $\boldsymbol{f}_{\boldsymbol{x}_{\mathrm{r}},\boldsymbol{\zeta}} : \mathcal{Z} \to \mathcal{X}$ *und* $\boldsymbol{f}_{\boldsymbol{\zeta}} : \mathcal{Z} \to \mathcal{Z}$ *mit*

$$\boldsymbol{x}_{\mathrm{r},k} = \boldsymbol{f}_{\boldsymbol{x}_{\mathrm{r}},\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k) \tag{3.6}$$

und

$$\boldsymbol{\zeta}_{k+1} = \boldsymbol{f}_{\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k) \tag{3.7}$$

charakterisiert, wobei $\boldsymbol{\zeta} \in \mathcal{Z} \subset \mathbb{R}^{n_\zeta}$, $n_\zeta < \infty$.

Basierend auf Definition 3.1 lässt sich die folgende formale Aussage formulieren[39].

Proposition 3.1
Die Solltrajektoriendarstellung sei nach Definition 3.1 ADP-kompatibel. Für ein durch die zeitdiskrete Systemdynamik (2.1) *beschriebenes System gilt dann:*

1. *Sind die durch*
$$\sum_{\kappa=k}^{\infty} \gamma^{\kappa-k} r\big(\boldsymbol{x}_\kappa, \boldsymbol{x}_{\mathrm{r},\kappa}, \boldsymbol{\mu}(\boldsymbol{x}_\kappa, \boldsymbol{\zeta}_\kappa)\big), \tag{3.8}$$
$0 < \gamma \leq 1$, *gegebenen Gesamtkosten (vgl.* (2.3a)*), die von der Sollzustandstrajektorie* $\boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{x}_{\mathrm{r},k+1}, \ldots$ *abhängen, endlich*[40]*, dann können sie durch eine Value Function der Form* $V^{\boldsymbol{\mu}}(\boldsymbol{x}_k, \boldsymbol{\zeta}_k)$ *beschrieben werden.*

2. *Sind die durch*
$$r(\boldsymbol{x}_k, \boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{u}_k) + \sum_{\kappa=k+1}^{\infty} \gamma^{\kappa-k} r\big(\boldsymbol{x}_\kappa, \boldsymbol{x}_{\mathrm{r},\kappa}, \boldsymbol{\mu}(\boldsymbol{x}_\kappa, \boldsymbol{\zeta}_\kappa)\big), \tag{3.9}$$
$0 < \gamma \leq 1$, *gegebenen Gesamtkosten (vgl.* (2.7)*), welche von der Sollzustandstrajektorie* $\boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{x}_{\mathrm{r},k+1}, \ldots$ *abhängen, endlich, dann können sie durch eine Q-Function der Form* $Q^{\boldsymbol{\mu}}(\boldsymbol{x}_k, \boldsymbol{\zeta}_k, \boldsymbol{u}_k)$ *beschrieben werden.*

[39] Ebenso gilt die Aussage von Proposition 3.1 auch für allgemeinere nichtlineare Systeme der Form $\boldsymbol{x}_{k+1} = \boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{u}_k)$, aus Gründen der Einheitlichkeit wurde hier jedoch der eingangsaffine Fall formuliert.

[40] Für sinnvoll gestellte Probleme schließt dies neben der Stabilisierbarkeit des Systems und einem zulässigen, d. h. insbesondere stabilisierenden, Regelgesetz $\boldsymbol{\mu}$ auch ein, dass durch die Wahl der Diskontierung γ bzw. des Solltrajektorienverlaufs die Gesamtkosten endlich bleiben.

Beweis:
Unter Verwendung von (2.1), (3.6) und (3.7) folgt

1. aus (3.8)

$$
\begin{aligned}
V^{\boldsymbol{\mu}}(\boldsymbol{x}_k, \boldsymbol{\zeta}_k) &\coloneqq \sum_{\kappa=k}^{\infty} \gamma^{\kappa-k} r\big(\boldsymbol{x}_\kappa, \boldsymbol{x}_{\mathrm{r},\kappa}, \boldsymbol{\mu}(\boldsymbol{x}_\kappa, \boldsymbol{\zeta}_\kappa)\big) && \text{(3.10a)}\\
&= r\big(\boldsymbol{x}_k, \boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{\mu}(\boldsymbol{x}_k, \boldsymbol{x}_{\mathrm{r},k})\big) + \gamma V^{\boldsymbol{\mu}}(\boldsymbol{x}_{k+1}, \boldsymbol{\zeta}_{k+1}) && \text{(3.10b)}\\
&= r\Big(\boldsymbol{x}_k, \boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k), \boldsymbol{\mu}\big(\boldsymbol{x}_k, \boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k)\big)\Big)\\
&\quad + \gamma V^{\boldsymbol{\mu}}\Big(\boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{g}(\boldsymbol{x}_k)\boldsymbol{\mu}\big(\boldsymbol{x}_k, \boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k)\big), \boldsymbol{f}_{\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k)\Big) && \text{(3.10c)}
\end{aligned}
$$

sowie

2. basierend auf (3.9) und analog zu (2.7) mit (3.10c)

$$
\begin{aligned}
Q^{\boldsymbol{\mu}}(\boldsymbol{x}_k, \boldsymbol{\zeta}_k, \boldsymbol{u}_k) &\coloneqq r(\boldsymbol{x}_k, \boldsymbol{x}_\mathrm{r}, \boldsymbol{u}_k) + \gamma V^{\boldsymbol{\mu}}(\boldsymbol{x}_{k+1}, \boldsymbol{\zeta}_{k+1}) && \text{(3.11a)}\\
&= r\big(\boldsymbol{x}_k, \boldsymbol{f}_{\boldsymbol{x}_{\mathrm{r},k},\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k), \boldsymbol{u}_k\big)\\
&\quad + \gamma V^{\boldsymbol{\mu}}\Big(\boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{g}(\boldsymbol{x}_k)\boldsymbol{\mu}\big(\boldsymbol{x}_k, \boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k)\big), \boldsymbol{f}_{\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k)\Big) && \text{(3.11b)}\\
&= r\big(\boldsymbol{x}_k, \boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k), \boldsymbol{u}_k\big)\\
&\quad + \gamma Q^{\boldsymbol{\mu}}\Big(\boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{g}(\boldsymbol{x}_k)\boldsymbol{\mu}\big(\boldsymbol{x}_k, \boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k)\big), \boldsymbol{f}_{\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k), \boldsymbol{\mu}\big(\boldsymbol{x}_k, \boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k)\big)\Big). && \text{(3.11c)}
\end{aligned}
$$

□

Eine ADP-kompatible Solltrajektoriendarstellung stellt somit sicher, dass die Repräsentation der Gesamtkosten in Form einer Value Function lediglich eine explizite Abhängigkeit von $\boldsymbol{x}_k$ und $\boldsymbol{\zeta}_k$ erfordert[41]. Zudem hängt eine Q-Function in diesem Fall nur explizit von $\boldsymbol{x}_k$, $\boldsymbol{\zeta}_k$ und $\boldsymbol{u}_k$ ab.

In der Arbeit von Yu et al. [YSH+17], bei der für jede neue Kombination aus Referenztrajektorie und Anfangszustand der Regler neu trainiert werden muss, ist die Kompatibilitätsforderung nach Definition 3.1 nicht erfüllt. Die Betrachtung der in den Abschnitten 2.2.2 und 2.2.3 diskutierten Methoden offenbart zwar, dass die dortigen zeitdiskreten Ansätze die Anforderung der ADP-Kompatibilität erfüllen[42]. Den existierenden Methoden gelingt es hierbei jedoch nicht, zu

[41] An dieser Stelle sei noch zu bemerken, dass auch Referenztrajektorienparameter, deren Folgezustand aus einer endlichen Anzahl vergangener Parameter hervorgeht, nach Definition 3.1 kompatibel sind. Um dies zu verdeutlichen, werde exemplarisch $\bar{\boldsymbol{\zeta}}_{k+1} = \bar{\boldsymbol{f}}(\bar{\boldsymbol{\zeta}}_k, \bar{\boldsymbol{\zeta}}_{k-1})$ mit $\bar{\boldsymbol{\zeta}}_k \in \bar{\mathcal{Z}} \subset \mathbb{R}^{n_{\bar{\zeta}}}$ betrachtet. Für den Referenzparameter $\boldsymbol{\zeta}_k \coloneqq [\bar{\boldsymbol{\zeta}}_k^\mathsf{T} \quad \bar{\boldsymbol{\zeta}}_{k-1}^\mathsf{T}]^\mathsf{T}$ gilt mit $\boldsymbol{f}(\boldsymbol{\zeta}_k) \coloneqq \bar{\boldsymbol{f}}(\bar{\boldsymbol{\zeta}}_k, \bar{\boldsymbol{\zeta}}_{k-1})$ aufgrund von $\boldsymbol{\zeta}_{k+1} = [\bar{\boldsymbol{f}}^\mathsf{T}(\bar{\boldsymbol{\zeta}}_k, \bar{\boldsymbol{\zeta}}_{k-1}) \quad \bar{\boldsymbol{\zeta}}_k^\mathsf{T}]^\mathsf{T} = \Big[\boldsymbol{f}^\mathsf{T}(\boldsymbol{\zeta}_k) \quad \Big(\Big[\boldsymbol{I}_{n_{\bar{\zeta}}} \quad \boldsymbol{0}_{n_{\bar{\zeta}} \times n_{\bar{\zeta}}}\Big]\boldsymbol{\zeta}_k\Big)^\mathsf{T}\Big]^\mathsf{T} \eqqcolon \boldsymbol{f}_{\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k)$ die Markov-Eigenschaft.

[42] Bei Ansätzen, bei denen ein stationärer Sollzustand vorgegeben wird, sind sowohl $\boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}(\cdot)$ als auch $\boldsymbol{f}_{\boldsymbol{\zeta}}(\cdot)$ Identitätsabbildungen.

generalisieren, indem eine flexible, veränderliche, externe Vorgabe des Solltrajektorienverlaufs, wie in Abbildung 2.4 visualisiert, berücksichtigt werden kann.

Bemerkung 3.1

Zwar wird im Zeitschritt k angenommen, dass die durch $\boldsymbol{\zeta}_k$, $\boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}(\cdot)$ und $\boldsymbol{f}_{\boldsymbol{\zeta}}(\cdot)$ beschriebene Solltrajektorie bis ins Unendliche fortgesetzt wird, jedoch kann aufgrund der expliziten Abhängigkeit der Value Function von $\boldsymbol{\zeta}_k$ dieser Parameter, der den Verlauf der Solltrajektorie definiert, prinzipiell jederzeit geändert werden. Wenngleich in jedem Zeitschritt k ein unendlicher Optimierungshorizont zugrunde gelegt wird, kann somit durch eine Anpassung von $\boldsymbol{\zeta}_k$ eine lokale Beschreibung der Solltrajektorie, d. h. ein im Zeitschritt k auf den lokalen Verlauf der Referenztrajektorie angepasster Parameter $\boldsymbol{\zeta}_k$, verwendet werden. Jedoch muss bei einem datenbasierten Training eines solchen optimalen Trajektorienfolgereglers $\boldsymbol{\mu}(\boldsymbol{x}_k, \boldsymbol{\zeta}_k)$ darauf geachtet werden, dass dem ADP-Algorithmus Datentupel

$$\mathcal{T}_k := \left\{\boldsymbol{x}_k, \boldsymbol{\zeta}_k, \boldsymbol{\mu}(\boldsymbol{x}_k, \boldsymbol{\zeta}_k), r(\boldsymbol{x}_k, \boldsymbol{\mu}(\boldsymbol{x}_k, \boldsymbol{\zeta}_k)), \boldsymbol{x}_{k+1}, \boldsymbol{\zeta}_{k+1}\right\} \tag{3.12}$$

(vgl. (3.3)) präsentiert werden, welche nach Definition 3.1 zu kompatiblen Solltrajektorien gehören. Dies erlaubt, neben der impliziten Abhängigkeit der Value Function $V^{\boldsymbol{\mu}}(\boldsymbol{x}_k, \boldsymbol{\zeta}_k)$ von der Diskontierung γ, der Systemdynamik $\boldsymbol{f}(\cdot)$, $\boldsymbol{g}(\cdot)$ und der Einschrittkostenfunktion $r(\cdot)$, auch die implizite Abhängigkeit vom Solltrajektorienverlauf, beschrieben durch $\boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}(\cdot)$ und $\boldsymbol{f}_{\boldsymbol{\zeta}}(\cdot)$, zu erlernen.

Die im vorliegenden Abschnitt erstmals formal definierte zeitdiskrete ADP-kompatible Solltrajektoriendarstellung liefert den ersten Beitrag zu der in Abschnitt 2.4.1 formulierten Forschungsfrage 1 und legt den Grundstein zur Entwicklung ADP-basierter Solltrajektorienfolgeregler.

Im weiteren Verlauf von Kapitel 3 werden Methoden vorgestellt, die extern vorgebbare Solltrajektorienverläufe durch nach Definition 3.1 kompatible Referenzverläufe entweder lokal approximieren oder auf einem endlichen Vorausschauhorizont nutzen und dabei, wie in Bemerkung 3.1 beschrieben, die Konsistenz der während des Adaptionsvorgangs verwendeten Daten gewährleisten (vgl. Abbildung 3.1). Nachfolgend werden zwei zeitdiskrete ADP-basierte Solltrajektorienregler präsentiert. Die zentrale Idee beider Methoden ist dabei, eine Q-Function zu definieren, die explizit den zukünftigen Verlauf der Solltrajektorie berücksichtigt. Bei der in Abschnitt 3.2 vorgestellten Methode geschieht dies mithilfe von Parametern, welche den Verlauf repräsentieren, wohingegen der Ansatz in Abschnitt 3.3 die Sollzustände über einen gleitenden Vorausschauhorizont explizit in die Q-Function integrieren.

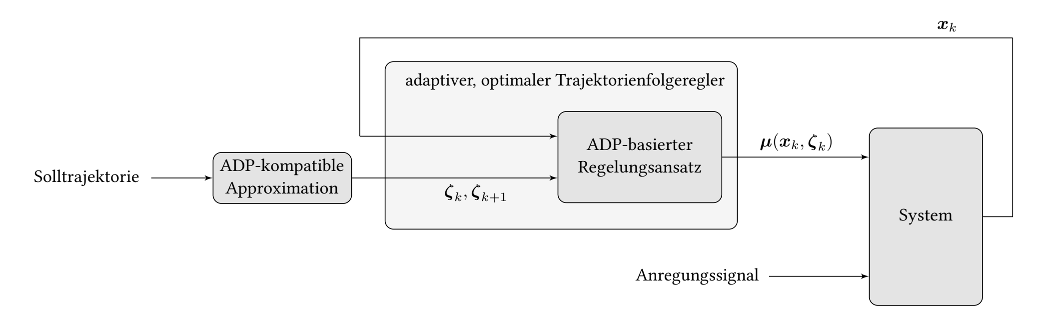


Abbildung 3.1: Aufgabe der ADP-kompatiblen Approximation einer Solltrajektorie ist es, aus einem im Zeitschritt k von außen vorgegebenen Referenztrajektorienverlauf eine nach Definition 3.1 kompatible Darstellung mit endlichdimensionalem Referenzparameter $\boldsymbol{\zeta}_k$ zu erzeugen. Aus den Tupeln $\{\boldsymbol{\zeta}_k, \boldsymbol{\zeta}_{k+1}\}$, die einem ADP-basierten Regelungsansatz zur Verfügung gestellt werden, kann dieser implizit $\boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}(\cdot)$ und $\boldsymbol{f}_{\boldsymbol{\zeta}}(\cdot)$ und somit den lokal im Zeitschritt k approximierten Solltrajektorienverlauf berücksichtigen.

3.2 Zeitdiskrete ADP-kompatible parametrierte Referenztrajektorie

Um im Zeitschritt k eine kompakte Darstellung des Einflusses des zukünftigen Solltrajektorienverlaufs auf die Gesamtkosten zu erhalten, wird im Folgenden eine Parametermatrix $\boldsymbol{Z}_k$ definiert, die Basisfunktionen $\boldsymbol{\rho}(\cdot)$ gewichtet und den Referenztrajektorienverlauf $\boldsymbol{x}_\mathrm{r}$ beschreibt. Diese Parametermatrix wird in die Q-Function $Q^{\boldsymbol{\mu}}(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k)$ integriert, hierbei entspricht $\boldsymbol{\zeta}_k = \mathrm{vec}(\boldsymbol{Z}_k)$[43]. Dadurch repräsentiert die Q-Function explizit die Abhängigkeit der akkumulierten (ggf. diskontierten) Gesamtkosten von dem durch $\boldsymbol{Z}_k$ beschriebenen und jederzeit beeinflussbaren Solltrajektorienverlauf. Der dieser Q-Function zugehörige Regler hängt letztlich ebenfalls explizit von der durch $\boldsymbol{Z}_k$ parametrierten Beschreibung der Referenztrajektorie ab, ein gelernter Regler muss bei einer Änderung des Solltrajektorienverlaufs also nicht erneut trainiert werden. Zudem wird im Optimierungsproblem nicht nur die aktuelle Abweichung vom Sollzustand, sondern der (parametrierte) Sollzustandsverlauf, wie beispielsweise Straßen-, Geschwindigkeits-, Konzentrations- oder Temperaturprofile, berücksichtigt. Dies erlaubt eine flexible Vorausschau von ADP-Reglern und verspricht Vorteile gegenüber einer konstanten Sollwertvorgabe. Nachfolgend wird dieser Ansatz als *Parametrized Reference ADP* (PRADP) bezeichnet.[44]

Nach Einführung dieser referenzabhängigen Q-Function können Funktionsapproximatoren verwendet werden, um basierend auf dem TD-Fehler sowohl die Q-Function als auch den optimalen Regler zu schätzen. Der im Folgenden präsentierte Ansatz stellt insbesondere aufgrund der speziellen Formulierung der solltrajektorienabhängigen Bellman-Gleichung unter Verwendung einer verschobenen Parametermatrix sicher, dass trotz beliebiger Vorgabe des

[43] Hierbei bezeichnet $\mathrm{vec}(\cdot)$ die Vektorisierung einer Matrix, indem die Spalten vertikal konkateniert werden.
[44] Teile des vorliegenden Abschnitts wurden in [KRP+20] veröffentlicht.

Solltrajektorienparameters ADP-Kompatibilität innerhalb der Trainingsdaten gewährleistet ist. Für den wichtigen Fall linear-quadratischer optimaler Trajektorienfolgeregler wird die dem Trajektorienfolgeregelungsproblem zugrunde liegende Q-Function anschließend analysiert. Unter plausiblen Annahmen werden Existenz und Eindeutigkeit der optimalen Lösung des betrachteten Optimierungsproblems bewiesen. Zudem wird die Stabilität des geschlossenen Regelkreises für die optimale Lösung bei Verwendung einer diskontierten Gütefunktion untersucht und ein hinreichendes Stabilitätskriterium vorgestellt. Anschließend werden Simulationsergebnisse, welche die Flexibilität der neuartigen Methode offenbaren, präsentiert und diskutiert. Für die reale Anwendung der in diesem Kapitel entwickelten ADP-kompatiblen Solltrajektorienbeschreibung sei auf Kapitel 6 verwiesen.

3.2.1 Allgemeine Problemstellung

Betrachtet werde zunächst ein zeitdiskretes System mit unbekannter Systemdynamik[45]

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k) + \boldsymbol{g}(\boldsymbol{x}_k)\boldsymbol{u}_k \tag{3.13}$$

mit dem diskreten Zeitschritt $k \in \mathbb{N}_{\geq 0}$, dem Systemzustand $\boldsymbol{x}_k \in \mathbb{R}^n$ und der Stellgröße $\boldsymbol{u}_k \in \mathbb{R}^p$. Das System (3.13) sei steuerbar auf der kompakten Menge $\mathcal{X} \subset \mathbb{R}^n$, die den Ursprung enthält (vgl. [KL15]). Weiterhin sei die durch die Parametermatrix

$$\boldsymbol{Z}_k = \begin{bmatrix} \boldsymbol{z}_{k,1}^\mathsf{T} \\ \boldsymbol{z}_{k,2}^\mathsf{T} \\ \vdots \\ \boldsymbol{z}_{k,n}^\mathsf{T} \end{bmatrix}, \tag{3.14}$$

$\boldsymbol{Z}_k \in \mathbb{R}^{n \times n_\mathrm{z}}$, und gegebene Basisfunktionen $\boldsymbol{\rho}(\kappa) \in \mathbb{R}^{n_\mathrm{z}}$ beschriebene Sollzustandstrajektorie $\boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k, \kappa) \in \mathbb{R}^n$ im Zeitschritt k durch

$$\boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k, \kappa) = \boldsymbol{Z}_k \boldsymbol{\rho}(\kappa) \tag{3.15}$$

definiert. Dabei beschreibt $\kappa \in \mathbb{N}_{\geq 0}$ den Zeitschritt auf der Solltrajektorie aus lokaler Perspektive des Zeitschritts k. Für $\kappa = 0$ ergibt sich somit der Sollzustand im Zeitschritt k, wohingegen für $\kappa > 0$ eine Vorausschau des Sollzustands für zukünftige Zeiten resultiert.

Das Ziel der optimalen Trajektorienfolgeregelung ist es, dass der Systemzustand $\boldsymbol{x}_{k+\kappa}$ dem Solltrajektorienverlauf $\boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k, \kappa)$, $\kappa = 0, 1, \ldots$, optimal bezüglich eines Gütefunktionals J_k folgt. Konkret soll für ein System mit unbekannter Systemdynamik ein Regler $\boldsymbol{\mu}^*(\boldsymbol{x}_k, \boldsymbol{Z}_k)$ gefunden werden, der das Gütefunktional

$$J_k = \sum_{\kappa=0}^{\infty} \gamma^\kappa r\big(\boldsymbol{x}_{k+\kappa}, \boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k, \kappa), \boldsymbol{u}_{k+\kappa}\big) \tag{3.16}$$

45 Grundsätzlich lässt sich der PRADP-Mechanismus auch auf allgemeine nichtlineare Systeme $\boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ anwenden, jedoch erfolgt dann für die Optimierung im Policy-Improvement-Schritt im Allgemeinen kein direkter analytischer Ausdruck mehr.

minimiert. Dabei beschreibt $\gamma \in (0,1]$ einen gegebenen Diskontierungsfaktor und $r(\cdot)$ nicht-negative Einschrittkosten, die beispielsweise Abweichungen des Systemzustands $\boldsymbol{x}_{k+\kappa}$ von $\boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k, \kappa)$ sowie die aufgebrachte Stellenergie bestrafen können. Somit ergibt sich die nachfolgende Problemstellung.

Problem 3.1
Für eine durch $\boldsymbol{Z}_k$ parametrierte Sollzustandstrajektorie nach (3.15) *werde die optimale Stellgrößensequenz, die das Kostenfunktional J_k gemäß* (3.16) *minimiert, mit $\boldsymbol{u}_k^*, \boldsymbol{u}_{k+1}^*, \dots$ und die damit verknüpften Kosten mit J_k^* bezeichnet. Zudem sei die Systemdynamik unbekannt. Gesucht ist die optimale Stellgröße $\boldsymbol{u}_k^* = \boldsymbol{\mu}^*(\boldsymbol{x}_k, \boldsymbol{Z}_k)$ in jedem Zeitschritt k in Abhängigkeit vom aktuellen Zustand $\boldsymbol{x}_k$ und der aktuellen Solltrajektorienbeschreibung $\boldsymbol{Z}_k$.*

3.2.2 Q-Function mit parametrierter Referenzdarstellung

Im Folgenden wird eine modifizierte Q-Function vorgestellt, deren minimierendes Regelgesetz eine Lösung $\boldsymbol{\mu}^*(\boldsymbol{x}_k, \boldsymbol{Z}_k)$ für Problem 3.1 darstellt. Diese Q-Function kann anschließend durch lineare Funktionsapproximatoren beschrieben und mithilfe einer ADP-Methode – beispielsweise dem LSPI-Ansatz [LP03] – ohne Kenntnis des Systemmodells aus Daten gelernt werden.

Um die durch J_k beschriebenen Kosten in (3.16) zu minimieren, muss die relative Zeit κ auf der aktuellen Solltrajektorie, die nach (3.15) durch $\boldsymbol{Z}_k$ parametriert ist, berücksichtigt werden. Dies geschieht durch die Verwendung einer um κ Zeitschritte verschobenen Parametermatrix $\boldsymbol{Z}_k^{(\kappa)}$ nach der folgenden Definition.

Definition 3.2 (Verschobene Parametermatrix $\boldsymbol{Z}_k^{(\kappa)}$)
Die Matrix $\boldsymbol{Z}_k^{(\kappa)}$ sei derart definiert, dass

$$\boldsymbol{x}_\mathrm{r}\left(\boldsymbol{Z}_k^{(\kappa)}, j\right) = \boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k, \kappa + j) \tag{3.17a}$$

$$\Leftrightarrow \boldsymbol{Z}_k^{(\kappa)} \boldsymbol{\rho}(j) = \boldsymbol{Z}_k \boldsymbol{\rho}(\kappa + j) \tag{3.17b}$$

gelte. Demnach ist

$$\boldsymbol{Z}_k^{(\kappa)} = \boldsymbol{Z}_k \boldsymbol{D}(\kappa) = \begin{bmatrix} \boldsymbol{z}_{k,1}^{(\kappa)\mathsf{T}} \\ \vdots \\ \boldsymbol{z}_{k,n}^{(\kappa)\mathsf{T}} \end{bmatrix} \tag{3.18}$$

eine modifizierte Version von $\boldsymbol{Z}_k = \boldsymbol{Z}_k^{(0)}$, sodass die zugehörige Referenztrajektorie um κ Zeitschritte verschoben ist, wobei $\boldsymbol{D}(\kappa)$ eine geeignete Matrix ist, sodass (3.17) *gilt.*

Anzumerken sei an dieser Stelle, dass $\boldsymbol{D}(\kappa)$ gemäß Definition 3.2 im Allgemeinen mehrdeutig ist, da für den Fall $n_\mathrm{z} > 1$ das durch (3.17b) gegebene Gleichungssystem, das verwendet wird, um $\boldsymbol{Z}_k^{(\kappa)}$ zu ermitteln, unterbestimmt ist.

Weiterhin gelte die Kurzschreibweise $\boldsymbol{x}_\mathrm{r}\left(\boldsymbol{Z}_k^{(\kappa)}\right) := \boldsymbol{x}_\mathrm{r}\left(\boldsymbol{Z}_k^{(\kappa)}, 0\right) = \boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k, \kappa)$. Mit $\boldsymbol{Z}_k^{(\kappa)}$ nach Definition 3.2 sei die erweiterte Q-Function, die den Solltrajektorienverlauf, parametriert durch $\boldsymbol{Z}_k$, explizit einbezieht, wie folgt definiert.

Definition 3.3 (Q-Function mit parametriertem Solltrajektorienverlauf)
Sei

$$\begin{aligned} Q^{\boldsymbol{\mu}}\left(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k\right) &:= r\left(\boldsymbol{x}_k, \boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k), \boldsymbol{u}_k\right) \\ &\quad + \sum_{\kappa=1}^{\infty} \gamma^\kappa r\left(\boldsymbol{x}_{k+\kappa}, \boldsymbol{x}_\mathrm{r}\left(\boldsymbol{Z}_k^{(\kappa)}\right), \boldsymbol{\mu}\left(\boldsymbol{x}_{k+\kappa}, \boldsymbol{Z}_k^{(\kappa)}\right)\right) \\ &= r\left(\boldsymbol{x}_k, \boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k), \boldsymbol{u}_k\right) \\ &\quad + \gamma Q^{\boldsymbol{\mu}}\left(\boldsymbol{x}_{k+1}, \boldsymbol{Z}_k^{(1)}, \boldsymbol{\mu}\left(\boldsymbol{x}_{k+1}, \boldsymbol{Z}_k^{(1)}\right)\right), \end{aligned} \tag{3.19}$$

wobei $\boldsymbol{\mu} : \mathbb{R}^n \times \mathbb{R}^{n \times n_\mathrm{z}} \to \mathbb{R}^p$ *das zu bewertende Regelgesetz bezeichne.*

Mit dieser Definition repräsentiert $Q^{\boldsymbol{\mu}}(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k)$ die akkumulierten (diskontierten) Kosten, wenn sich das System im Zustand $\boldsymbol{x}_k$ befindet, die Stellgröße $\boldsymbol{u}_k$ im Zeitschritt k angewandt wird, danach dem Regelgesetz $\boldsymbol{\mu}(\cdot)$ gefolgt wird, und die Sollzustandstrajektorie durch $\boldsymbol{Z}_k$ parametriert ist. Basierend auf (3.19) ergibt sich die optimale Q-Function zu

$$\begin{aligned} Q^*\left(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k\right) &= r\left(\boldsymbol{x}_k, \boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k), \boldsymbol{u}_k\right) \\ &\quad + \min_{\boldsymbol{\mu}} \gamma Q^{\boldsymbol{\mu}}\left(\boldsymbol{x}_{k+1}, \boldsymbol{Z}_k^{(1)}, \boldsymbol{\mu}\left(\boldsymbol{x}_{k+1}, \boldsymbol{Z}_k^{(1)}\right)\right) \\ &= r\left(\boldsymbol{x}_k, \boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k), \boldsymbol{u}_k\right) \\ &\quad + \gamma Q^*\left(\boldsymbol{x}_{k+1}, \boldsymbol{Z}_k^{(1)}, \boldsymbol{\mu}^*\left(\boldsymbol{x}_{k+1}, \boldsymbol{Z}_k^{(1)}\right)\right). \end{aligned} \tag{3.20}$$

Dabei bezeichnet $\boldsymbol{\mu}^*(\cdot)$ das optimale Regelgesetz, d. h. es gilt $\boldsymbol{\mu}^*\left(\boldsymbol{x}_{k+1}, \boldsymbol{Z}_k^{(1)}\right) = \boldsymbol{u}_{k+1}^*$. Wie das nachfolgende Lemma, das die Zusammenhänge des klassischen Q-Learnings [WD92] auf den PRADP-Fall überträgt, zeigt, erweist sich diese Q-Function als hilfreich, um Problem 3.1 zu lösen.

Lemma 3.1
Die Stellgröße $\boldsymbol{u}_k$*, die* $Q^*\left(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k\right)$ *minimiert, stellt eine Lösung für* $\boldsymbol{u}_k^*$ *nach Problem 3.1 dar.*

Beweis:
Mit (3.20) folgt

$$\begin{aligned}\min_{\boldsymbol{u}_k} Q^*\left(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k\right) &= r\left(\boldsymbol{x}_k, \boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k), \boldsymbol{u}_k^*\right) + \gamma Q^*\left(\boldsymbol{x}_{k+1}, \boldsymbol{Z}_k^{(1)}, \boldsymbol{u}_{k+1}^*\right) \\ &= \min_{\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \dots} \sum_{\kappa=0}^{\infty} \gamma^\kappa r\left(\boldsymbol{x}_\kappa, \boldsymbol{x}_\mathrm{r}\left(\boldsymbol{Z}_k^{(\kappa)}\right), \boldsymbol{u}_\kappa\right) \\ &= J_k^*\end{aligned} \tag{3.21}$$

und somit direkt die Aussage des Lemmas. □

Wenn die optimale Q-Function $Q^*(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k)$ bekannt ist, ergibt sich nach Lemma 3.1 die gesuchte optimale Stellgröße $\boldsymbol{u}_k^*$ durch

$$\boldsymbol{u}_k^* = \arg\min_{\boldsymbol{u}_k} Q^*(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k). \tag{3.22}$$

Im nächsten Abschnitt wird die Q-Function durch lineare Funktionsapproximatoren beschrieben und basierend auf dem TD-Fehler die unbekannte Q-Function geschätzt.

3.2.3 Funktionsapproximation und Policy Iteration der erweiterten Q-Function

Da klassisches, tabellarisches Q-Learning wertkontinuierliche Zustands- und Stellgrößenräume nicht angemessen handhaben kann [van12], wird die als stetig angenommene Q-Function durch einen linearen Funktionsapproximator

$$Q^*\left(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k\right) = \boldsymbol{w}^{*\mathsf{T}} \boldsymbol{\phi}\left(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k\right) + \epsilon\left(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k\right) \tag{3.23}$$

beschrieben. Hierbei bezeichnet $\boldsymbol{w}^* \in \mathbb{R}^h$ den unbekannten optimalen Gewichtsvektor, $\boldsymbol{\phi} \in \mathbb{R}^h$ einen geeigneten Basisfunktionsvektor und ϵ den Approximationsfehler (vgl. Abschnitt 2.1.3). Da $\boldsymbol{w}^*$ a priori unbekannt ist, sei die geschätzte optimale Q-Function durch

$$\hat{Q}^*\left(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k\right) = \hat{\boldsymbol{w}}^\mathsf{T} \boldsymbol{\phi}\left(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k\right) \tag{3.24}$$

gegeben. Analog zu (3.22) folgt daraus das geschätzte optimale Regelgesetz

$$\hat{\boldsymbol{\mu}}^*(\boldsymbol{x}_k, \boldsymbol{Z}_k) = \arg\min_{\boldsymbol{u}_k} \hat{Q}^*\left(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k\right). \tag{3.25}$$

Basierend auf der parametrierten Q-Function ergibt sich der zugehörige TD-Fehler [Sut88] wie nachfolgend definiert.

Definition 3.4 (TD-Fehler der erweiterten Q-Function)
Der TD-Fehler, der sich mit der geschätzten Q-Function $\hat{Q}^*(\cdot)$ (3.24) *aus der Bellman-Gleichung* (3.20) *ergibt, sei durch*

$$\begin{aligned}\delta_k &:= r\left(\boldsymbol{x}_k, \boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k), \boldsymbol{u}_k\right) + \gamma \hat{Q}^*\left(\boldsymbol{x}_{k+1}, \boldsymbol{Z}_k^{(1)}, \hat{\boldsymbol{\mu}}^*\left(\boldsymbol{x}_{k+1}, \boldsymbol{Z}_k^{(1)}\right)\right) \\ &\quad - \hat{Q}^*\left(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k\right) \\ &= r\left(\boldsymbol{x}_k, \boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k), \boldsymbol{u}_k\right) + \gamma \hat{\boldsymbol{w}}^\intercal \boldsymbol{\phi}\left(\boldsymbol{x}_{k+1}, \boldsymbol{Z}_k^{(1)}, \hat{\boldsymbol{\mu}}^*\left(\boldsymbol{x}_{k+1}, \boldsymbol{Z}_k^{(1)}\right)\right) \\ &\quad - \hat{\boldsymbol{w}}^\intercal \boldsymbol{\phi}\left(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k\right)\end{aligned} \tag{3.26}$$

definiert.

Da der TD-Fehler die Qualität der Approximation der Q-Function quantifiziert, wird das Gewicht $\hat{\boldsymbol{w}} \in \mathbb{R}^h$ gesucht, welches den quadratischen TD-Fehler δ_k^2 minimiert. Da (3.26) eine skalare Gleichung darstellt, werden hierzu $M \geq h$ Tupel

$$\mathcal{T}_k := \left\{r_k, \hat{Q}_k^*, \hat{Q}_k^{*+}\right\}, \quad k = 1, \ldots, M, \tag{3.27a}$$

mit

$$r_k := r\left(\boldsymbol{x}_k, \boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k), \boldsymbol{u}_k\right), \tag{3.27b}$$

$$\hat{Q}_k^* := \hat{\boldsymbol{w}}^\intercal \boldsymbol{\phi}_k := \hat{\boldsymbol{w}}^\intercal \boldsymbol{\phi}\left(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k\right), \tag{3.27c}$$

$$\hat{Q}_k^{*+} := \hat{\boldsymbol{w}}^\intercal \boldsymbol{\phi}_k^+ := \hat{\boldsymbol{w}}^\intercal \boldsymbol{\phi}\left(\boldsymbol{x}_{k+1}, \boldsymbol{Z}_k^{(1)}, \hat{\boldsymbol{\mu}}^*\left(\boldsymbol{x}_{k+1}, \boldsymbol{Z}_k^{(1)}\right)\right) \tag{3.27d}$$

verwendet. Diese M Tupel werden aus gemessenen oder simulierten Systemtrajektorien $(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{x}_{k+1})$, den Solltrajektorienparametern $\boldsymbol{Z}_k$ und verschobenen Solltrajektorienparametern $\boldsymbol{Z}_k^{(1)}$ gebildet. Anschließend kann $\hat{\boldsymbol{w}}$ mit einer geeigneten ADP-Methode, beispielsweise mit dem LSPI-Algorithmus [LP03], geschätzt werden. Mit den Tupeln $\mathcal{T}_k$, $k = 1, \ldots, M$, folgt aus (3.26)[46]

$$\underbrace{\begin{bmatrix}\delta_1 \\ \vdots \\ \delta_M\end{bmatrix}}_{=:\boldsymbol{\delta}} = \underbrace{\begin{bmatrix}r_1 \\ \vdots \\ r_M\end{bmatrix}}_{=:\boldsymbol{r}} + \underbrace{\left(\gamma \begin{bmatrix}\boldsymbol{\phi}_1^{+\intercal} \\ \vdots \\ \boldsymbol{\phi}_M^{+\intercal}\end{bmatrix} - \begin{bmatrix}\boldsymbol{\phi}_1^\intercal \\ \vdots \\ \boldsymbol{\phi}_M^\intercal\end{bmatrix}\right)}_{=:\boldsymbol{\Phi}} \hat{\boldsymbol{w}}. \tag{3.28}$$

Sofern die Anregungsbedingung

$$\mathrm{Rang}\left(\boldsymbol{\Phi}^\intercal \boldsymbol{\Phi}\right) = h \tag{3.29}$$

erfüllt ist, existiert nach Åström und Wittenmark [ÅW95, Theorem 2.1] durch

$$\hat{\boldsymbol{w}} = \left(\boldsymbol{\Phi}^\intercal \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^\intercal \boldsymbol{r} \tag{3.30}$$

eine eindeutige Lösung $\hat{\boldsymbol{w}}$, welche $\boldsymbol{\delta}^\intercal \boldsymbol{\delta}$ minimiert.

[46] Dies stellt im Wesentlichen eine Erweiterung von [LP03, Abschnitt 5.1] auf ADP-Solltrajektorienfolgeregler mit parametriertem Referenzverlauf dar.

Bemerkung 3.2

Die Verwendung von $\boldsymbol{Z}_k^{(1)} = \boldsymbol{Z}_k \boldsymbol{D}(1)$ *(vgl.* (3.18)*) anstelle von beliebigen nachfolgenden Parametern* $\boldsymbol{Z}_{k+1}$ *in den Trainingstupeln* $\mathcal{T}_k$ (3.27a) *ist essenziell, da hierdurch (in Kombination mit der Systemdynamik* (3.13)*) gewährleistet wird, dass die für ADP-Methoden wichtige Markov-Eigenschaft erfüllt ist (vgl. Abschnitt 3.1 und [LV09]). Insbesondere wird hierdurch sichergestellt, dass die für den ADP-Formalismus verwendete Solltrajektoriendarstellung ADP-kompatibel nach Definition 3.1 ist, denn mit*[47]

$$\boldsymbol{\zeta}_k := \operatorname{vec}(\boldsymbol{Z}_k) \tag{3.31}$$

folgt in Analogie zu (3.6)

$$\boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k) := \boldsymbol{Z}_k \boldsymbol{\rho}(0) = \operatorname{mat}(\boldsymbol{\zeta}_k, n, n_z)\boldsymbol{\rho}(0) \tag{3.32}$$

(vgl. (3.15)*) und analog zu* (3.7)

$$\boldsymbol{f}_{\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k) := \operatorname{vec}\Big(\boldsymbol{Z}_k^{(1)}\Big) = \operatorname{vec}\big(\boldsymbol{Z}_k \boldsymbol{D}(1)\big) = \operatorname{vec}\big(\operatorname{mat}(\boldsymbol{\zeta}_k, n, n_z)\boldsymbol{D}(1)\big). \tag{3.33}$$

An dieser Stelle sei anzumerken, dass $\hat{\boldsymbol{\mu}}^*(\cdot)$, welches in $\hat{Q}_k^{*+}$ nach (3.27d) verwendet wird, selbst eine Schätzung (für das optimale Regelgesetz) darstellt. Dieser Mechanismus, dass eine Schätzung basierend auf einer anderen Schätzung erfolgt, ist im Reinforcement Learning unter dem Begriff *bootstrapping* bekannt (vgl. [SB18]). Daher genügt eine einmalige Schätzung von $\hat{\boldsymbol{w}}$ basierend auf der Least-Squares-Lösung von (3.30) nicht, um die optimale Q-Function und den optimalen Regler zu ermitteln. Stattdessen wird eine Policy Iteration, beginnend bei einem zulässigen Initialgewicht $\hat{\boldsymbol{w}}^{[0]}$, durchgeführt. Dieses Vorgehen ist in Algorithmus 3.1 zusammengefasst, wobei der Schwellwert $e_{\hat{\boldsymbol{w}}}$ eine Abbruchbedingung darstellt.

Algorithmus 3.1 PRADP mithilfe des LSPI-Algorithmus

1: **Initialisiere** $l := 0, \hat{\boldsymbol{w}}^{[0]}$, sodass $\hat{\boldsymbol{\mu}}^{[0]}(\cdot)$ zulässig ist
2: **do**
3: Policy Evaluation: berechne $\hat{\boldsymbol{w}}^{[l+1]}$ basierend auf (3.30) mit $\hat{\boldsymbol{w}} = \hat{\boldsymbol{w}}^{[l+1]}$
4: Policy Improvement: berechne $\hat{\boldsymbol{\mu}}^{[l+1]}$ nach (3.25)
5: $l := l + 1$
6: **while** $\left\|\hat{\boldsymbol{w}}^{[l]} - \hat{\boldsymbol{w}}^{[l-1]}\right\|_2 > e_{\hat{\boldsymbol{w}}}$

[47] Hierbei bildet $\operatorname{mat}(\cdot)$ aus einem Vektor eine Matrix, wobei $\operatorname{mat}(\operatorname{vec}(\boldsymbol{M}), \tilde{n}, p) = \boldsymbol{M} \in \mathbb{R}^{\tilde{n} \times p}$ gilt.

Bemerkung 3.3

Aufgrund der Verwendung einer Q-Function, die explizit von der grundsätzlich beliebigen Stellgröße $\boldsymbol{u}_k$ *abhängt, liegt ein Off-Policy-Verfahren vor (vgl. Abschnitt 2.1.4.4). Dies erweist sich insbesondere als vorteilhaft, da während des Aufzeichnens der* M *Datentupel* $\mathcal{T}_k$, $k = 1, \ldots, M$, *üblicherweise eine Anregung des Systems stattfinden muss, um die durch* (3.29) *gegebene Anregungsbedingung zu erfüllen und einen erfolgreichen Policy-Evaluation-Schritt zu gewährleisten. Die Behavior Policy, d. h. die Stellgröße* $\boldsymbol{u}_k$, *die tatsächlich auf das System angewandt wird, kann daher zur Systemanregung verwendet werden (beispielsweise durch additive Überlagerung von Rauschen oder harmonischen Schwingungen, siehe Kapitel 5). Demgegenüber wird die Target Policy* $\hat{\boldsymbol{\mu}}^*$, *die dem geschätzten optimalen Regelgesetz nach* (3.25) *entspricht, im Ausdruck* $\gamma \hat{Q}^* \left(\boldsymbol{x}_{k+1}, \boldsymbol{Z}_k^{(1)}, \hat{\boldsymbol{\mu}}^* \left(\boldsymbol{x}_{k+1}, \boldsymbol{Z}_k^{(1)} \right) \right)$ *in* (3.26) *verwendet, weshalb die mit dem geschätzten optimalen Regelgesetz assoziierte Q-Function gelernt wird.*

Mit $\hat{Q}^{[l]}(\cdot) = \hat{\boldsymbol{w}}^{[l]\mathsf{T}} \boldsymbol{\phi}(\cdot)$ und $Q^{\hat{\boldsymbol{\mu}}^{[l]}}$ nach (3.19), wobei $\boldsymbol{\mu} = \hat{\boldsymbol{\mu}}^{[l]}$ sei, lässt sich die in [LP03, Theorem 7.1] gegebene Konvergenzaussage des LSPI-Algorithmus direkt auf den auf parametrierte Solltrajektorienverläufe erweiterten Fall übertragen.

Proposition 3.2 (Konvergenz der Q-Function nach [LP03, Theorem 7.1])

Sei für alle Iterationen l *durch* $\bar{\epsilon} \geq 0$ *eine obere Schranke für den Approximationsfehler zwischen der zum Regelgesetz* $\hat{\boldsymbol{\mu}}^{[l]}$ *gehörenden geschätzten Q-Function* $\hat{Q}^{[l]}$ *und der wahren Q-Function* $Q^{\hat{\boldsymbol{\mu}}^{[l]}}$ *gegeben, d. h.*

$$\left\| \hat{Q}^{[l]} - Q^{\hat{\boldsymbol{\mu}}^{[l]}} \right\|_\infty \leq \bar{\epsilon}, \quad \forall l = 1, 2, \ldots. \tag{3.34}$$

Dann gilt für Algorithmus 3.1:

$$\limsup_{l \to \infty} \left\| \hat{Q}^{[l]} - Q^* \right\|_\infty \leq \frac{2\gamma\bar{\epsilon}}{(1-\gamma)^2}. \tag{3.35}$$

Beweis:

Der Beweis ist in [LP03, Theorem 7.1] bzw. [BT96, Proposition 6.2] zu finden. □

Lagoudakis und Parr [LP03] betonen, dass eine geeignete Wahl der Basisfunktionen $\boldsymbol{\phi}(\cdot)$ sowie die Erzeugung der Datentupel (d. h. eine geeignete Anregung) maßgeblich für die Fehlerschranke $\bar{\epsilon}$ verantwortlich sind. Nach Proposition 3.2 konvergiert Algorithmus 3.1 für adäquate Funktionsapproximatoren $\boldsymbol{\phi}(\cdot)$ sowie eine angemessene Anregung somit gegen eine Nachbarschaft der optimalen erweiterten Q-Function. Jedoch ist für eingangsaffine Systeme (3.13) bzw. generelle nichtlineare Systeme $\boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ und allgemeine Kostenfunktionale (3.16) eine geeignete

Wahl an Basisfunktionen sowie deren Anzahl nach wie vor ein ungelöstes Problem[48] [WHL17]. Zudem erfordert eine geeignete Systemanregung im allgemeinen, nichtlinearen Fall, dass die Trainingsdaten alle relevanten Bereiche des Zustands- und Stellgrößenraums abdecken, um die vorhandenen Nichtlinearitäten angemessen zu berücksichtigen und optimale Gewichte $\boldsymbol{w}^*$ lernen zu können. Die neue ADP-basierte Solltrajektorienfolgeregelungsmethode wird im Folgenden anhand linearer Systeme und quadratischer Gütefunktionale betrachtet. Dieser in zahlreichen regelungstechnischen Problemen relevante LQ-Fall ermöglicht analytische Einsichten in die Struktur der Q-Function $Q^*\left(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k\right)$ und somit eine geeignete Wahl der Basisfunktionen $\boldsymbol{\phi}(\cdot)$ zur Funktionsapproximation.

3.2.4 Linear-quadratische optimale Trajektorienfolgeregelung mit parametrierter Referenz

Im Folgenden sei die Systemdynamik durch

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \tag{3.36}$$

und das Kostenfunktional durch

$$\begin{aligned} J_k &= \sum_{\kappa=0}^{\infty} \gamma^\kappa \left[\left(\boldsymbol{x}_{k+\kappa} - \boldsymbol{x}_\mathrm{r}\left(\boldsymbol{Z}_k^{(\kappa)}\right)\right)^\intercal \boldsymbol{Q}\left(\boldsymbol{x}_{k+\kappa} - \boldsymbol{x}_\mathrm{r}\left(\boldsymbol{Z}_k^{(\kappa)}\right)\right) + \boldsymbol{u}_{k+\kappa}^\intercal \boldsymbol{R}\boldsymbol{u}_{k+\kappa}\right] \\ &=: \sum_{\kappa=0}^{\infty} \gamma^\kappa \left[\boldsymbol{e}_{k,\kappa}^\intercal \boldsymbol{Q}\boldsymbol{e}_{k,\kappa} + \boldsymbol{u}_{k+\kappa}^\intercal \boldsymbol{R}\boldsymbol{u}_{k+\kappa}\right] \end{aligned} \tag{3.37}$$

mit $\boldsymbol{e}_{k,\kappa} \coloneqq \boldsymbol{x}_{k+\kappa} - \boldsymbol{x}_\mathrm{r}\left(\boldsymbol{Z}_k^{(\kappa)}\right)$ beschrieben. Hierbei bestraft $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ die Abweichung des Systemzustands $\boldsymbol{x}_{k+\kappa}$ vom Sollzustand $\boldsymbol{x}_\mathrm{r}\left(\boldsymbol{Z}_k^{(\kappa)}\right)$ und $\boldsymbol{R} \in \mathbb{R}^{p\times p}$ gewichtet den Stellaufwand. Zudem seien die folgenden Annahmen erfüllt.

Annahme 3.1
Sei $\boldsymbol{Q} = \boldsymbol{Q}^\intercal \succeq \boldsymbol{0}$, $\boldsymbol{R} = \boldsymbol{R}^\intercal \succ \boldsymbol{0}$, $(\boldsymbol{A}, \boldsymbol{B})$ steuerbar und $(\boldsymbol{A}, \boldsymbol{C})$ detektierbar, wobei $\boldsymbol{C}$ derart definiert ist, dass $\boldsymbol{C}^\intercal\boldsymbol{C} = \boldsymbol{Q}$ gilt[49].

Annahme 3.2
Sei die Matrix $\boldsymbol{D}(\kappa)$, welche die verschobene Parametermatrix $\boldsymbol{Z}_k^{(\kappa)}$ nach (3.18) *definiert (vgl. Definition 3.2), so, dass $|\lambda_j| < 1, \forall j = 1, \ldots, n_\mathrm{z}$, gilt, wobei λ_j die Eigenwerte von $\sqrt{\gamma}\boldsymbol{D}(1)$ sind.*

48 Wang et al. [WHL17] bezeichnen die geeignete Wahl des Funktionsapproximators als „more of an art than science“.

49 Jede symmetrische, positiv semidefinite Matrix lässt sich als $\boldsymbol{Q} = \boldsymbol{C}^\intercal\boldsymbol{C}$ schreiben, beispielsweise durch Diagonalisierung (vgl. [Lib12, S. 195]).

Die Bedeutung dieser Annahmen wird in der folgenden Bemerkung eingeordnet.

Bemerkung 3.4

Annahme 3.1 ist üblich für linear-quadratische Optimalregelungsprobleme. Hierdurch wird die Existenz und Eindeutigkeit einer stabilisierenden Lösung der zeitdiskreten algebraischen Riccati-Gleichung des Regulationsproblems, das durch (3.36) *und* (3.37) *für* $\boldsymbol{Z}_k = \mathbf{0}$ *(d. h. für einen Sollzustand* $\mathbf{0}$ *für alle* k*) gegeben ist, gewährleistet (vgl. [Kuč72, Theorem 8]).*
Zudem ist es einleuchtend, dass für eine sinnvolle Problemstellung der Solltrajektorienverlauf $\boldsymbol{x}_\mathrm{r}\left(\boldsymbol{Z}_k^{(\kappa)}\right)$ *derart definiert sein muss, dass ein Regelgesetz existiert, das mit einem endlichen Gütefunktional* J_k *verknüpft ist. Wie der nachfolgende Satz 3.1 zeigen wird, garantieren Annahme 3.1 und Annahme 3.2 die Existenz einer solchen Lösung.*

Bevor das optimale Regelgesetz hergeleitet wird, wird der Trajektorienfehler $\boldsymbol{e}_{k,\kappa}$ aus (3.37) durch

$$\begin{aligned}
\boldsymbol{e}_{k,\kappa} &= \boldsymbol{x}_{k+\kappa} - \boldsymbol{x}_\mathrm{r}\left(\boldsymbol{Z}_k^{(\kappa)}\right) \\
&= \boldsymbol{x}_{k+\kappa} - \boldsymbol{Z}_k^{(\kappa)}\boldsymbol{\rho}(0) \\
&= \underbrace{\begin{bmatrix} \boldsymbol{I}_n & \begin{bmatrix} -\boldsymbol{\rho}(0) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & -\boldsymbol{\rho}(0) \end{bmatrix}^\mathsf{T} \end{bmatrix}}_{=:\boldsymbol{M}} \underbrace{\begin{bmatrix} \boldsymbol{x}_{k+\kappa} \\ \boldsymbol{z}_{k,1}^{(\kappa)} \\ \vdots \\ \boldsymbol{z}_{k,n}^{(\kappa)} \end{bmatrix}}_{=:\tilde{\boldsymbol{x}}_{k,\kappa}},
\end{aligned} \tag{3.38}$$

$\kappa = 0, 1, \ldots,$ ausgedrückt. $\boldsymbol{I}_n$ bezeichnet die $n \times n$-Einheitsmatrix und $\tilde{\boldsymbol{x}}_{k,\kappa}$ stellt den um den Referenzparameter $\boldsymbol{Z}_k^{(\kappa)}$ erweiterten Systemzustand $\boldsymbol{x}_{k+\kappa}$ dar. Der zugehörige Optimalregler ist durch den folgenden Satz gegeben.

Satz 3.1 (Optimales Regelgesetz des zeitdiskreten parametrischen Trajektorienfolgeregelungsproblems)
Seien eine Sollzustandsrepräsentation $\boldsymbol{x}_\mathrm{r}\left(\boldsymbol{Z}_k^{(\kappa)}\right)$ *(vgl.* (3.15)*) und eine Verschiebungsmatrix* $\boldsymbol{D}(\kappa)$ *wie in Definition 3.2 gegeben.*

1. *Das optimale Regelgesetz, das* (3.37) *unter Berücksichtigung der Systemdynamik* (3.36) *minimiert, ist linear bezüglich* $\tilde{\boldsymbol{x}}_{k,\kappa}$ *in* (3.38) *und somit durch*

$$\boldsymbol{\mu}^*\left(\boldsymbol{x}_{k+\kappa}, \boldsymbol{Z}_k^{(\kappa)}\right) = \boldsymbol{u}_{k+\kappa}^* = -\boldsymbol{K}^*\tilde{\boldsymbol{x}}_{k,\kappa}, \quad \kappa = 0, 1, \dots, \tag{3.39}$$

gegeben. Die optimale Verstärkungsmatrix $\boldsymbol{K}^*$ *ergibt sich zu*

$$\boldsymbol{K}^* = (\gamma\tilde{\boldsymbol{B}}^\mathsf{T}\tilde{\boldsymbol{P}}\tilde{\boldsymbol{B}} + \boldsymbol{R})^{-1}\gamma\tilde{\boldsymbol{B}}^\mathsf{T}\tilde{\boldsymbol{P}}\tilde{\boldsymbol{A}}, \tag{3.40}$$

mit

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}^\mathsf{T}(1) & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{D}^\mathsf{T}(1) \end{bmatrix}, \quad \tilde{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{bmatrix}, \tag{3.41}$$

$\tilde{\boldsymbol{A}} \in \mathbb{R}^{n(n_\mathrm{z}+1)\times n(n_\mathrm{z}+1)}$, $\tilde{\boldsymbol{B}} \in \mathbb{R}^{n(n_\mathrm{z}+1)\times p}$. *In* (3.40) *bezeichnet* $\tilde{\boldsymbol{P}}$ *die Lösung der zeitdiskreten algebraischen Riccati-Gleichung*

$$\tilde{\boldsymbol{P}} = \gamma\tilde{\boldsymbol{A}}^\mathsf{T}\tilde{\boldsymbol{P}}\tilde{\boldsymbol{A}} - \gamma\tilde{\boldsymbol{A}}^\mathsf{T}\tilde{\boldsymbol{P}}\tilde{\boldsymbol{B}}(\boldsymbol{R} + \tilde{\boldsymbol{B}}^\mathsf{T}\tilde{\boldsymbol{P}}\tilde{\boldsymbol{B}})^{-1}\tilde{\boldsymbol{B}}^\mathsf{T}\tilde{\boldsymbol{P}}\tilde{\boldsymbol{A}} + \tilde{\boldsymbol{Q}} \tag{3.42}$$

mit $\tilde{\boldsymbol{Q}} = \boldsymbol{M}^\mathsf{T}\boldsymbol{Q}\boldsymbol{M}$ *und* $\boldsymbol{M}$ *nach* (3.38).

2. *Des Weiteren sind unter Annahme 3.1 und Annahme 3.2 Existenz und Eindeutigkeit des optimalen Regelgesetzes* $\boldsymbol{\mu}^*\left(\boldsymbol{x}_{k+\kappa}, \boldsymbol{Z}_k^{(\kappa)}\right)$ *gesichert.*

Beweis:

1. Mit (3.38) kann das Gütefunktional (3.37) zu

$$J_k = \sum_{\kappa=0}^{\infty} \gamma^\kappa \left[\tilde{\boldsymbol{x}}_{k,\kappa}^\mathsf{T}\boldsymbol{M}^\mathsf{T}\boldsymbol{Q}\boldsymbol{M}\tilde{\boldsymbol{x}}_{k,\kappa} + \boldsymbol{u}_{k+\kappa}^\mathsf{T}\boldsymbol{R}\boldsymbol{u}_{k+\kappa}\right] \tag{3.43}$$

umgeformt werden. Mit (3.36) und (3.18) gilt zudem

$$\tilde{\boldsymbol{x}}_{k,\kappa+1} = \begin{bmatrix} \boldsymbol{A}\boldsymbol{x}_{k+\kappa} + \boldsymbol{B}\boldsymbol{u}_{k+\kappa} \\ \boldsymbol{D}(1)^\mathsf{T}\boldsymbol{z}_{k,1}^{(\kappa)} \\ \vdots \\ \boldsymbol{D}(1)^\mathsf{T}\boldsymbol{z}_{k,n}^{(\kappa)} \end{bmatrix} = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}_{k,\kappa} + \tilde{\boldsymbol{B}}\boldsymbol{u}_{k+\kappa}. \tag{3.44}$$

Mit γ, $\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{B}}$, $\tilde{\boldsymbol{Q}}$ und $\boldsymbol{R}$ ergibt sich somit ein *diskontiertes* Standard-LQ-Optimalregelungsproblem aus (3.43) und (3.44) für den erweiterten Zustand $\tilde{\boldsymbol{x}}_{k,\kappa}$. Dieses diskontierte Problem ist äquivalent zum undiskontierten LQ-Optimalregelungsproblem mit $\sqrt{\gamma}\tilde{\boldsymbol{A}}$, $\sqrt{\gamma}\tilde{\boldsymbol{B}}$, $\tilde{\boldsymbol{Q}}$ und $\boldsymbol{R}$ (vgl. [GGH+18]). Für dieses undiskontierte Standard-LQ-Optimalregelungsproblem ist wohlbekannt, dass der optimale Regler linear bezüglich des Zustands (d. h. hier des erweiterten Zustands $\tilde{\boldsymbol{x}}_{k,\kappa}$) ist und die optimale Zustandsrückführungsmatrix durch (3.40) gegeben ist [LVS12, Abschnitt 2.4]. Daher gilt (3.39) und somit die erste Aussage von Satz 3.1.

2. Bezüglich der zweiten Aussage sei zunächst angemerkt, dass die Stabilisierbarkeit von $(\sqrt{\gamma}\tilde{\boldsymbol{A}}, \sqrt{\gamma}\tilde{\boldsymbol{B}})$ direkt aus Annahme 3.1 und Annahme 3.2 folgt, da $(\boldsymbol{A}, \boldsymbol{B})$ steuerbar ist und $|\lambda_j| < 1, \forall j = 1, \ldots, n_\mathrm{z}$, gilt. Zudem folgt aus $\boldsymbol{Q} \succeq \boldsymbol{0}$ direkt $\tilde{\boldsymbol{Q}} \succeq \boldsymbol{0}$. Weil $(\boldsymbol{A}, \boldsymbol{C})$ nach Annahme 3.1 detektierbar ist und da alle zusätzlichen Zustände in $\tilde{\boldsymbol{A}}$ im Vergleich zu $\boldsymbol{A}$ aufgrund von Annahme 3.2 stabil sind, folgt, dass $(\sqrt{\gamma}\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{C}})$ (mit $\tilde{\boldsymbol{C}}$ so, dass $\tilde{\boldsymbol{C}}^\mathsf{T}\tilde{\boldsymbol{C}} = \tilde{\boldsymbol{Q}}$) ebenfalls detektierbar ist. Schließlich folgt nach Kučera [Kuč72, Theorem 8], dass wegen $\tilde{\boldsymbol{Q}} \succeq \boldsymbol{0}$, $\boldsymbol{R} \succ \boldsymbol{0}$, $(\sqrt{\gamma}\tilde{\boldsymbol{A}}, \sqrt{\gamma}\tilde{\boldsymbol{B}})$ stabilisierbar und $(\sqrt{\gamma}\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{C}})$ detektierbar eine eindeutige Lösung existiert. □

Bemerkung 3.5

Satz 3.1 offenbart, dass im Fall bekannter Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ die optimale Lösung $\boldsymbol{K}^*$ direkt durch das Lösen der zeitdiskreten algebraischen Riccati-Gleichung (vgl. [AL84]), die zu $\sqrt{\gamma}\tilde{\boldsymbol{A}}$, $\sqrt{\gamma}\tilde{\boldsymbol{B}}$, $\tilde{\boldsymbol{Q}}$ und $\boldsymbol{R}$ gehört, resultiert.

Gleichung (3.44) zeigt letztlich auch, dass die, aus ADP-Sicht fundamentale, Markov-Eigenschaft gilt. Ausgehend von Satz 3.1 wird für den Fall unbekannter Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ die folgende LQ-PRADP-Problemstellung formuliert.

Problem 3.2 (LQ-PRADP)
Sei die durch die Matrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ beschriebene Systemdynamik unbekannt. Gesucht ist die Reglermatrix $\boldsymbol{K}^*$ (vgl. (3.39)), die das Kostenfunktional J_k (3.37) minimiert. Die optimale Stellgröße im Zeitschritt k ist dabei durch

$$\boldsymbol{u}_k^* = \boldsymbol{\mu}^*\left(\boldsymbol{x}_k, \boldsymbol{Z}_k^{(0)}\right) = -\boldsymbol{K}^*\tilde{\boldsymbol{x}}_{k,0} =: -\begin{bmatrix}\boldsymbol{K}_\mathrm{x}^* & \boldsymbol{K}_\mathrm{Z}^*\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_k \\ \boldsymbol{z}_{k,1}^{(0)} \\ \vdots \\ \boldsymbol{z}_{k,n}^{(0)}\end{bmatrix} \tag{3.45}$$

gegeben.

Bevor das optimale Regelgesetz $\boldsymbol{K}^*$ ohne Kenntnis der Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ mit einem ADP-Ansatz gelernt wird, wird im folgenden Lemma zunächst die genaue Struktur der optimalen Q-Function $Q^*\left(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k\right)$, die zu Problem 3.2 gehört, analysiert.

Lemma 3.2 (Struktur der Q-Function mit parametrierter Solltrajektoriendarstellung)
Die zu Problem 3.2 gehörende Q-Function weist die quadratische Form

$$Q^*(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k) = \bar{\boldsymbol{y}}_k^\mathsf{T}\boldsymbol{H}\bar{\boldsymbol{y}}_k = \begin{bmatrix}\boldsymbol{x}_k \\ \boldsymbol{u}_k \\ \boldsymbol{z}_{k,1:n}\end{bmatrix}^\mathsf{T}\begin{bmatrix}\boldsymbol{h}_\mathrm{xx} & \boldsymbol{h}_\mathrm{xu} & \boldsymbol{h}_\mathrm{xz} \\ \boldsymbol{h}_\mathrm{ux} & \boldsymbol{h}_\mathrm{uu} & \boldsymbol{h}_\mathrm{uz} \\ \boldsymbol{h}_\mathrm{zx} & \boldsymbol{h}_\mathrm{zu} & \boldsymbol{h}_\mathrm{zz}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_k \\ \boldsymbol{u}_k \\ \boldsymbol{z}_{k,1:n}\end{bmatrix} \tag{3.46}$$

mit $\bar{\boldsymbol{y}}_k := \begin{bmatrix}\boldsymbol{x}_k^\mathsf{T} & \boldsymbol{u}_k^\mathsf{T} & \boldsymbol{z}_{k,1:n}^\mathsf{T}\end{bmatrix}^\mathsf{T} := \begin{bmatrix}\boldsymbol{x}_k^\mathsf{T} & \boldsymbol{u}_k^\mathsf{T} & \boldsymbol{z}_{k,1}^\mathsf{T} & \dots & \boldsymbol{z}_{k,n}^\mathsf{T}\end{bmatrix}^\mathsf{T}$ auf, wobei $\boldsymbol{H}$ so gewählt sei, dass $\boldsymbol{H} = \boldsymbol{H}^\mathsf{T}$ gilt.

Beweis:
Mit (3.19) und (3.20) folgt

$$\begin{aligned} Q^*(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k) = {} & r\left(\boldsymbol{x}_k, \boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k), \boldsymbol{u}_k\right) \\ & + \sum_{\kappa=1}^{\infty}\gamma^\kappa r\left(\boldsymbol{x}_{k+\kappa}, \boldsymbol{x}_\mathrm{r}\left(\boldsymbol{Z}_k^{(\kappa)}\right), \boldsymbol{\mu}^*\left(\boldsymbol{x}_{k+\kappa}, \boldsymbol{Z}_k^{(\kappa)}\right)\right). \end{aligned} \tag{3.47}$$

Wegen (3.36), (3.39) und (3.18) folgt, dass die Zustände $\boldsymbol{x}_{k+\kappa}$ und Stellgrößen $\boldsymbol{\mu}^*\left(\boldsymbol{x}_{k+\kappa}, \boldsymbol{Z}_k^{(\kappa)}\right)$ $\forall\kappa = 0, 1, \dots$ linear bezüglich $\bar{\boldsymbol{y}}_k$ sind. Aus dieser linearen Abhängigkeit und mit (3.38) folgt Linearität von $\boldsymbol{e}_{k,\kappa}$ bezüglich $\bar{\boldsymbol{y}}_k$, $\forall\kappa \geq 0$. Durch die Linearität von $\boldsymbol{e}_{k,\kappa}$ und $\boldsymbol{\mu}^*(\cdot)$ bezüglich $\bar{\boldsymbol{y}}_k$ und die quadratische Struktur von $r(\cdot)$ in (3.37) ist die Q-Function in (3.47) quadratisch bezüglich $\bar{\boldsymbol{y}}_k$ und folglich gilt (3.46). □

Eine Konsequenz von Lemma 3.2 ist, dass die optimale Q-Function Q^* exakt durch einen Funktionsapproximator $\hat{Q}^*$ beschrieben werden kann, wenn $\hat{\boldsymbol{w}} = \boldsymbol{w}^*$ den nicht-redundanten Elementen der Matrix $\boldsymbol{H} = \boldsymbol{H}^\intercal$ entspricht[50] und die Funktionsapproximatoren zu

$$\boldsymbol{\phi}\left(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k\right) = \bar{\boldsymbol{y}}_k \otimes_{\mathrm{r}} \bar{\boldsymbol{y}}_k \tag{3.48}$$

gewählt werden[51]. Basierend auf Lemma 3.2 kann das optimale Regelgesetz wie nachfolgend gezeigt durch die Matrix $\boldsymbol{H}$ und ohne explizite Verwendung der Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ angegeben werden.

Satz 3.2 (Optimales Regelgesetz in Abhängigkeit von $\boldsymbol{H}$)
Die eindeutige erweiterte optimale Zustandsrückführung, die J_k (3.37) minimiert, ist durch

$$\boldsymbol{u}_k^* = \boldsymbol{\mu}^*(\boldsymbol{x}_k, \boldsymbol{Z}_k) = -\boldsymbol{K}^*\tilde{\boldsymbol{x}}_{k,0} = \underbrace{-\,\boldsymbol{h}_{\mathrm{uu}}^{-1}\begin{bmatrix}\boldsymbol{h}_{\mathrm{ux}} & \boldsymbol{h}_{\mathrm{uz}}\end{bmatrix}}_{\begin{bmatrix}\boldsymbol{K}_{\mathrm{x}}^* & \boldsymbol{K}_{\mathrm{Z}}^*\end{bmatrix}}\begin{bmatrix}\boldsymbol{x}_k \\ \boldsymbol{z}_{k,1:n}\end{bmatrix} \tag{3.49}$$

gegeben.

Beweis:
Nach Lemma 3.1 minimiert die Stellgröße $\boldsymbol{u}_k^*$, die $Q^*(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k)$ minimiert, auch J_k. Mit (3.46) und $\boldsymbol{H} = \boldsymbol{H}^\intercal$ liefert die notwendige Bedingung

$$\frac{\partial Q^*(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k)}{\partial \boldsymbol{u}_k} = 2\left(\boldsymbol{h}_{\mathrm{ux}}\boldsymbol{x}_k + \boldsymbol{h}_{\mathrm{uz}}\boldsymbol{z}_{k,1:n} + \boldsymbol{h}_{\mathrm{uu}}\boldsymbol{u}_k\right) \overset{!}{=} \boldsymbol{0} \tag{3.50}$$

die in (3.49) gegebene Stellgröße $\boldsymbol{u}_k^*$. Des Weiteren zeigt

$$\frac{\partial^2 Q^*(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k)}{\partial \boldsymbol{u}_k^2} = 2\boldsymbol{h}_{\mathrm{uu}}, \tag{3.51}$$

dass $\boldsymbol{h}_{\mathrm{uu}} \succ \boldsymbol{0}$ gelten muss, damit die Stellgröße $\boldsymbol{u}_k^*$ in (3.49) auch tatsächlich J_k in (3.37) minimiert. Daher wird im Folgenden $\boldsymbol{h}_{\mathrm{uu}} \succ \boldsymbol{0}$ gezeigt. Sei $Q_{\mathrm{reg}}^*(\boldsymbol{x}_k, \boldsymbol{u}_k)$ die optimale Q-Function, die den Regulationsfall mit $\boldsymbol{x}_{\mathrm{r}}\left(\boldsymbol{Z}_k^{(\kappa)}\right) = \boldsymbol{x}_{\mathrm{r}}(\boldsymbol{0}, \kappa) = \boldsymbol{0}$ beschreibt. Dann gilt

$$Q^*(\boldsymbol{x}_k, \boldsymbol{0}, \boldsymbol{u}_k) = Q_{\mathrm{reg}}^*(\boldsymbol{x}_k, \boldsymbol{u}_k), \quad \forall \boldsymbol{x}_k \in \mathbb{R}^n, \boldsymbol{u}_k \in \mathbb{R}^p. \tag{3.52}$$

Zudem gilt nach Bradtke et al. [BYB94]

$$\begin{aligned} Q_{\mathrm{reg}}^*(\boldsymbol{x}_k, \boldsymbol{u}_k) &= \begin{bmatrix}\boldsymbol{x}_k \\ \boldsymbol{u}_k\end{bmatrix}^\intercal \begin{bmatrix}\boldsymbol{h}_{\mathrm{reg,xx}} & \boldsymbol{h}_{\mathrm{reg,xu}} \\ \boldsymbol{h}_{\mathrm{reg,ux}} & \boldsymbol{h}_{\mathrm{reg,uu}}\end{bmatrix} \begin{bmatrix}\boldsymbol{x}_k \\ \boldsymbol{u}_k\end{bmatrix} \\ &= \begin{bmatrix}\boldsymbol{x}_k \\ \boldsymbol{u}_k\end{bmatrix}^\intercal \begin{bmatrix}\gamma\boldsymbol{A}^\intercal\boldsymbol{P}\boldsymbol{A} + \boldsymbol{Q} & \gamma\boldsymbol{A}^\intercal\boldsymbol{P}\boldsymbol{B} \\ \gamma\boldsymbol{B}^\intercal\boldsymbol{P}\boldsymbol{A} & \gamma\boldsymbol{B}^\intercal\boldsymbol{P}\boldsymbol{B} + \boldsymbol{R}\end{bmatrix} \begin{bmatrix}\boldsymbol{x}_k \\ \boldsymbol{u}_k\end{bmatrix} \end{aligned} \tag{3.53}$$

50 Nicht-redundant meint hierbei alle Elemente h_{ij} der symmetrischen Matrix $\boldsymbol{H}$ mit $i \leq j$. Elemente von $\hat{\boldsymbol{w}}$, die zu Nebendiagonalelementen von $\boldsymbol{H}$ gehören, werden zudem mit dem Faktor 2 multipliziert.

51 Der Operator $\otimes_{\mathrm{r}}$ berechnet für identische Operanden das reduzierte Kronecker-Produkt mit nicht-redundanten Elementen.

für den Regulationsfall. Hierbei ist $\boldsymbol{P}$ die Lösung der zeitdiskreten algebraischen Riccati-Gleichung

$$\boldsymbol{P} = \gamma \boldsymbol{A}^\intercal \boldsymbol{P} \boldsymbol{A} - \gamma \boldsymbol{A}^\intercal \boldsymbol{P} \boldsymbol{B} (\boldsymbol{R} + \boldsymbol{B}^\intercal \boldsymbol{P} \boldsymbol{B})^{-1} \boldsymbol{B}^\intercal \boldsymbol{P} \boldsymbol{A} + \boldsymbol{Q}. \tag{3.54}$$

Unter Annahme 3.1 existiert eine eindeutige Lösung $\boldsymbol{P} = \boldsymbol{P}^\intercal \succeq \boldsymbol{0}$ (vgl. [Kuč72, Theorem 8]). Daher folgt aus (3.52) und (3.53)

$$\boldsymbol{h}_{\text{uu}} = \boldsymbol{h}_{\text{reg,uu}} = \gamma \boldsymbol{B}^\intercal \boldsymbol{P} \boldsymbol{B} + \boldsymbol{R} \succ \boldsymbol{0} \tag{3.55}$$

und somit die gesuchte Aussage. □

Da in (3.16) potenziell ein diskontiertes Gütemaß verwendet wird und Satz 3.1 aufgrund der Analogie der diskontierten Problemstellung zum nicht-diskontierten Problem mit der Systemmatrix $\sqrt{\gamma}\tilde{\boldsymbol{A}}$ und der Eingangsmatrix $\sqrt{\gamma}\tilde{\boldsymbol{B}}$ zunächst nur Stabilität des fiktiven Systems

$$\tilde{\boldsymbol{x}}_{k+1} = \sqrt{\gamma}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}_k - \sqrt{\gamma}\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}^*\tilde{\boldsymbol{x}}_k \tag{3.56}$$

gewährleistet, muss nachfolgend noch überprüft werden, ob für die gewählte Diskontierung $0 < \gamma \leq 1$ das geregelte System

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k - \boldsymbol{B}\boldsymbol{K}_{\text{x}}^*\boldsymbol{x}_k - \boldsymbol{B}\boldsymbol{K}_{\zeta}^*\boldsymbol{\zeta}_k \tag{3.57}$$

für beschränkte Referenzparametervorgaben $\|\boldsymbol{\zeta}_k\|_2 < \infty$ mit $\boldsymbol{\zeta}_k = \text{vec}(\boldsymbol{Z}_k)$ stabil ist, d. h. ob sämtliche Eigenwerte von $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_{\text{x}}^*$ innerhalb des Einheitskreises liegen. In Anlehnung an die Eingangs-Zustands-Stabilität [Kha02, Definition 4.7] sei dieser Zusammenhang wie nachfolgend definiert als zeitdiskrete Referenz-Zustands-Stabilität bezeichnet.

Definition 3.5 (Zeitdiskrete Referenz-Zustands-Stabilität)
Ein geregeltes System ist Referenz-Zustands-stabil, wenn für beschränkte Referenzparametervorgaben

$$\|\boldsymbol{\zeta}_k\|_2 < \infty$$

mit $\boldsymbol{\zeta}_k = \text{vec}(\boldsymbol{Z}_k)$ *beschränkte Systemzustände*

$$\|\boldsymbol{x}_k\|_2 < \infty, \; \forall k \geq 0,$$

resultieren.

Bei Kenntnis von $\boldsymbol{A}$ und $\boldsymbol{B}$ können die Eigenwerte von $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_{\text{x}}^*$ mithilfe des berechneten oder gelernten Reglers $\boldsymbol{K}^* = \begin{bmatrix}\boldsymbol{K}_{\text{x}}^* & \boldsymbol{K}_{\text{Z}}^*\end{bmatrix}$ direkt überprüft werden. Da die durch $\boldsymbol{A}$ und $\boldsymbol{B}$ beschriebene Systemdynamik jedoch insbesondere im Fall adaptiver Optimalregler nicht immer bekannt ist, wird nachfolgend ein hinreichendes Kriterium gegeben, mit dem ausschließlich anhand von $\boldsymbol{Q}$, $\boldsymbol{R}$ und der ggf. mittels eines ADP-Ansatzes gelernten Q-Funktion $Q^*(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k)$ überprüft werden kann, ob die Reglermatrix $\boldsymbol{K}^*$ das System (3.57) stabilisiert.

Satz 3.3 (Stabilitätskriterium des Solltrajektorienfolgeregelungsproblems)
Sei $\boldsymbol{H}$ wie in (3.46) definiert. Aus $Q^*(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k)$ folgt $\boldsymbol{P}^* := \boldsymbol{h}_{\mathrm{xx}} - \boldsymbol{h}_{\mathrm{xu}}\boldsymbol{h}_{\mathrm{uu}}^{-1}\boldsymbol{h}_{\mathrm{ux}}$ und $\boldsymbol{K}_{\mathrm{x}}^* = \boldsymbol{h}_{\mathrm{uu}}^{-1}\boldsymbol{h}_{\mathrm{ux}}$ (vgl. (3.49)). Wenn

$$b_\gamma := \frac{1}{\lambda_{\min}\left(\boldsymbol{P}^*\left(\boldsymbol{P}^* - \boldsymbol{Q} - \boldsymbol{K}_{\mathrm{x}}^{*\mathsf{T}}\boldsymbol{R}\boldsymbol{K}_{\mathrm{x}}^*\right)^{-1}\right)} < \gamma \leq 1 \tag{3.58}$$

gilt, dann liegen sämtliche Eigenwerte von $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_{\mathrm{x}}^*$ innerhalb des Einheitskreises und das mit $\boldsymbol{K}^*$ geregelte System ist Referenz-Zustands-stabil nach Definition 3.5.

Beweis:
Nach Satz 3.2 und wegen (3.52) folgt, dass mit Kenntnis von $Q^*(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k)$ auch die optimale Q-Funktion $Q^*_{\mathrm{reg}}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ des Regulierungsproblems ohne Vorgabe einer Solltrajektorie bekannt ist. Wegen

$$\begin{aligned}
V^*(\boldsymbol{x}_k) &= Q^*_{\mathrm{reg}}(\boldsymbol{x}_k, \boldsymbol{\mu}^*(\boldsymbol{x}_k)) \\
&= \boldsymbol{x}_k^{\mathsf{T}}\boldsymbol{h}_{\mathrm{xx}}\boldsymbol{x}_k + 2\boldsymbol{x}_k^{\mathsf{T}}\boldsymbol{h}_{\mathrm{xu}}\boldsymbol{\mu}^*(\boldsymbol{x}_k) + \boldsymbol{\mu}^{*\mathsf{T}}(\boldsymbol{x}_k)\boldsymbol{h}_{\mathrm{uu}}\boldsymbol{\mu}^*(\boldsymbol{x}_k) \\
&= \boldsymbol{x}_k^{\mathsf{T}}\boldsymbol{h}_{\mathrm{xx}}\boldsymbol{x}_k - 2\boldsymbol{x}_k^{\mathsf{T}}\boldsymbol{h}_{\mathrm{xu}}\boldsymbol{h}_{\mathrm{uu}}^{-1}\boldsymbol{h}_{\mathrm{ux}}\boldsymbol{x}_k + \boldsymbol{x}_k^{\mathsf{T}}\boldsymbol{h}_{\mathrm{xu}}\boldsymbol{h}_{\mathrm{uu}}^{-1}\boldsymbol{h}_{\mathrm{uu}}\boldsymbol{h}_{\mathrm{uu}}^{-1}\boldsymbol{h}_{\mathrm{ux}}\boldsymbol{x}_k \\
&= \boldsymbol{x}_k^{\mathsf{T}}\underbrace{\left(\boldsymbol{h}_{\mathrm{xx}} - \boldsymbol{h}_{\mathrm{xu}}\boldsymbol{h}_{\mathrm{uu}}^{-1}\boldsymbol{h}_{\mathrm{ux}}\right)}_{=:\boldsymbol{P}^*}\boldsymbol{x}_k
\end{aligned} \tag{3.59}$$

stellt $\boldsymbol{P}^*$ die positiv definite Lösung der diskreten algebraischen Riccati-Gleichung (3.54) dar und das durch

$$\boldsymbol{x}_{k+1} = \underbrace{\sqrt{\gamma}\boldsymbol{A}}_{=:\boldsymbol{A}_\gamma}\boldsymbol{x}_k - \underbrace{\sqrt{\gamma}\boldsymbol{B}}_{=:\boldsymbol{B}_\gamma}\boldsymbol{K}_{\mathrm{x}}^*\boldsymbol{x}_k \tag{3.60}$$

gegebene System wird durch $\boldsymbol{K}_{\mathrm{x}}^*$ stabilisiert. Somit liegen alle Eigenwerte von $\boldsymbol{A}_\gamma - \boldsymbol{B}_\gamma\boldsymbol{K}_{\mathrm{x}}^*$ innerhalb des Einheitskreises. Im Folgenden wird überprüft, ob für ein gegebenes γ auch Stabilität des durch $\frac{1}{\sqrt{\gamma}}(\boldsymbol{A}_\gamma - \boldsymbol{B}_\gamma\boldsymbol{K}_{\mathrm{x}}^*) = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_{\mathrm{x}}^*$ beschriebenen geschlossenen Regelkreises gilt.

Hierzu werde zunächst untersucht, für welche Konstanten $c \geq 1$ das System

$$\boldsymbol{x}_{k+1} = c\,(\boldsymbol{A}_\gamma - \boldsymbol{B}_\gamma\boldsymbol{K}_{\mathrm{x}}^*)\,\boldsymbol{x}_k \tag{3.61}$$

stabil ist. Betrachtet werde dazu der Lyapunov-Kandidat $V^*(\boldsymbol{x}_k) = \boldsymbol{x}_k^{\mathsf{T}}\boldsymbol{P}^*\boldsymbol{x}_k$. Somit gilt

$$V^*(\boldsymbol{x}_{k+1}) - V^*(\boldsymbol{x}_k) = \boldsymbol{x}_k^{\mathsf{T}}\underbrace{\left((c\boldsymbol{A}_\gamma - c\boldsymbol{B}_\gamma\boldsymbol{K}_{\mathrm{x}}^*)^{\mathsf{T}}\,\boldsymbol{P}^*\,(c\boldsymbol{A}_\gamma - c\boldsymbol{B}_\gamma\boldsymbol{K}_{\mathrm{x}}^*) - \boldsymbol{P}^*\right)}_{=:\bar{\boldsymbol{M}}}\boldsymbol{x}_k \tag{3.62}$$

und (3.61) ist garantiert stabil, wenn $\bar{\boldsymbol{M}}$ negativ definit ist. Durch Umformung ergibt sich

$$\begin{aligned}
\bar{\boldsymbol{M}} &= c^2 \boldsymbol{A}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{A}_\gamma - \boldsymbol{P}^* - c^2 \boldsymbol{K}_\mathrm{x}^{*\intercal} \boldsymbol{B}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{A}_\gamma - c^2 \boldsymbol{A}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{B}_\gamma \boldsymbol{K}_\mathrm{x}^* + c^2 \boldsymbol{K}_\mathrm{x}^{*\intercal} \boldsymbol{B}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{B}_\gamma \boldsymbol{K}_\mathrm{x}^* \\
&\overset{(3.54)}{=} -c^2 \boldsymbol{Q} + c^2 \boldsymbol{P}^* - \boldsymbol{P}^* + c^2 \boldsymbol{A}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{B}_\gamma \left(\boldsymbol{B}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{B}_\gamma + \boldsymbol{R}\right)^{-1} \boldsymbol{B}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{A}_\gamma \\
&\quad - c^2 \boldsymbol{K}_\mathrm{x}^{*\intercal} \boldsymbol{B}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{A}_\gamma - c^2 \boldsymbol{A}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{B}_\gamma \boldsymbol{K}_\mathrm{x}^* + c^2 \boldsymbol{K}_\mathrm{x}^{*\intercal} \boldsymbol{B}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{B}_\gamma \boldsymbol{K}_\mathrm{x}^* \\
&= -c^2 \boldsymbol{Q} - (1-c^2)\boldsymbol{P}^* - c^2 \boldsymbol{K}_\mathrm{x}^{*\intercal} \boldsymbol{B}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{A}_\gamma + c^2 \boldsymbol{K}_\mathrm{x}^{*\intercal} \boldsymbol{B}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{B}_\gamma \boldsymbol{K}_\mathrm{x}^* \\
&= -c^2 \boldsymbol{Q} - (1-c^2)\boldsymbol{P}^* - c^2 \boldsymbol{K}_\mathrm{x}^{*\intercal} \left(\boldsymbol{B}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{B}_\gamma + \boldsymbol{R}\right) \boldsymbol{K}_\mathrm{x}^* + c^2 \boldsymbol{K}_\mathrm{x}^{*\intercal} \boldsymbol{B}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{B}_\gamma \boldsymbol{K}_\mathrm{x}^* \\
&= -c^2 \boldsymbol{Q} - (1-c^2)\boldsymbol{P}^* - c^2 \boldsymbol{K}_\mathrm{x}^{*\intercal} \boldsymbol{R} \boldsymbol{K}_\mathrm{x}^* \\
&= c^2 \left(\boldsymbol{P}^* - \boldsymbol{Q} - \boldsymbol{K}_\mathrm{x}^{*\intercal} \boldsymbol{R} \boldsymbol{K}_\mathrm{x}^*\right) - \boldsymbol{P}^* \overset{!}{\prec} \boldsymbol{0},
\end{aligned} \tag{3.63}$$

wobei die aus $\boldsymbol{K}_\mathrm{x}^* = \left(\boldsymbol{B}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{B}_\gamma + \boldsymbol{R}\right)^{-1} \boldsymbol{B}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{A}_\gamma$ resultierende Beziehung $\boldsymbol{B}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{A}_\gamma = \left(\boldsymbol{B}_\gamma^\intercal \boldsymbol{P}^* \boldsymbol{B}_\gamma + \boldsymbol{R}\right) \boldsymbol{K}_\mathrm{x}^*$ genutzt wird. Somit ist $\bar{\boldsymbol{M}}$ stets negativ definit, wenn

$$1 \le c^2 < \lambda_\mathrm{min}\left(\boldsymbol{P}^* \left(\boldsymbol{P}^* - \boldsymbol{Q} - \boldsymbol{K}_\mathrm{x}^{*\intercal} \boldsymbol{R} \boldsymbol{K}_\mathrm{x}^*\right)^{-1}\right) \tag{3.64}$$

gilt. Mit $c^2 = \frac{1}{\gamma}$ folgt schließlich die Aussage des Satzes. □

Das nachfolgende Beispiel verdeutlicht, dass Satz 3.3 ein hinreichendes Stabilitätskriterium für den geschlossenen Regelkreis $\boldsymbol{x}_{k+1} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_\mathrm{x}^*)\,\boldsymbol{x}_k$ liefert.

Beispiel 3.1 *Für das durch*

$$\boldsymbol{A} = \begin{bmatrix} 3 & 0 \\ 0 & 0{,}4 \end{bmatrix}, \qquad \boldsymbol{B} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix},$$
$$\boldsymbol{Q} = \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}, \qquad \boldsymbol{R} = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.65}$$

und unterschiedliche Diskontierungen γ gegebene Optimierungsproblem folgen die Zusammenhänge in Tabelle 3.1. Hierbei ist $\gamma > b_\gamma$ nach Satz 3.3 hinreichend für die Stabilität von $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_\mathrm{x}^*$, ohne $\boldsymbol{A}$ und $\boldsymbol{B}$ explizit zu kennen.

γ	$\lvert\mathrm{eig}\,(\boldsymbol{A}_\gamma - \boldsymbol{B}_\gamma \boldsymbol{K}_\mathrm{x}^*)\rvert$	$\lvert\mathrm{eig}\,(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_\mathrm{x}^*)\rvert$	b_γ	$\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_\mathrm{x}^*$ *stabil*	$\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_\mathrm{x}^*$ *stabil nach Satz 3.3*
1	0,4883; 0,0286	0,4883; 0,0286	0,4337	*ja*	*ja*
0,6	0,6055; 0,0354	0,7818; 0,0457	0,5498	*ja*	*ja*
0,5	0,6484; 0,0380	0,9170; 0,0537	0,5836	*ja*	*k. A.*[52]
0,4	0,6976; 0,0412	1,1031; 0,0651	0,6142	*nein*	*k. A.*

Tabelle 3.1: Eigenwerte des fiktiven Systems $\boldsymbol{A}_\gamma - \boldsymbol{B}_\gamma \boldsymbol{K}_\mathrm{x}^*$ sowie des realen Systems $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_\mathrm{x}^*$ bei Regelung mit $\boldsymbol{K}_\mathrm{x}^*$, welches aus der optimalen Lösung des mit γ diskontierten linear-quadratischen Optimierungsproblems mit den Parametern aus (3.65) resultiert. Die hinreichende untere Schranke b_γ folgt aus Satz 3.3.

Gemäß Satz 3.2 können, falls $\boldsymbol{H}$ (oder alternativ $\boldsymbol{w}^*$) bekannt ist, sowohl Q^* als auch $\boldsymbol{\mu}^*$ direkt berechnet werden. Deshalb ist das Ziel des PRADP-Ansatzes, $\boldsymbol{w}^*$ mithilfe von Datentupeln $\mathcal{T}_k$, $k = 1, \ldots, M$, (vgl. (3.27a)) zu bestimmen. Da nach Lemma 3.2 die optimale Q-Function $Q^*(\boldsymbol{x}_k, \boldsymbol{Z}_k, \boldsymbol{u}_k)$ zu Problem 3.2 quadratisch bezüglich $\bar{\boldsymbol{y}}_k$ ist, können hierfür quadratische Funktionsapproximatoren gewählt werden. Zudem liefert (3.49) in Satz 3.2 einen analytischen Zusammenhang zwischen $\boldsymbol{H}$ und dem optimalen Regelgesetz $\boldsymbol{\mu}^*(\boldsymbol{x}_k, \boldsymbol{Z}_k)$. Daher können $\boldsymbol{w}^*$ und somit auch der optimale Regler $\boldsymbol{\mu}^*(\boldsymbol{x}_k, \boldsymbol{Z}_k)$ datenbasiert und ohne explizite Kenntnis der Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ gelernt werden – beispielsweise durch Verwendung der LSPI in Algorithmus 3.1. Hierbei ergibt sich der Policy-Improvement-Schritt direkt zu

$$\hat{\boldsymbol{\mu}}^{[l+1]}(\boldsymbol{x}_k, \boldsymbol{Z}_k) = -\left(\boldsymbol{h}_\mathrm{uu}^{[l+1]}\right)^{-1} \begin{bmatrix} \boldsymbol{h}_\mathrm{ux}^{[l+1]} & \boldsymbol{h}_\mathrm{uz}^{[l+1]} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{z}_{k,1:n} \end{bmatrix}. \tag{3.66}$$

Eine konkrete ADP-Implementierung wird im Folgenden präsentiert. Zudem werden Simulationsergebnisse gezeigt und diskutiert.

3.2.5 Implementierung und simulativer Vergleich des PRADP

In diesem Abschnitt werden die Vorteile des vorgestellten PRADP-Verfahrens gegenüber einem weitverbreiteten Ansatz aus der Literatur, der annimmt, der Solltrajektorienverlauf folge global einer unbekannten Exosystemdynamik, simulativ gezeigt. Dazu wird zunächst beschrieben, wie ein beliebiger Solltrajektorienverlauf durch die beispielhafte und ADP-kompatible Wahl kubischer Polynome aus lokaler Sicht im Zeitschritt k approximiert werden kann. Anschließend wird ein Beispielproblem definiert. Für dieses Beispielproblem wird einerseits der in dieser Arbeit vorgestellte PRADP-Ansatz und andererseits die besagte Vergleichsmethode aus der Literatur trainiert. Für beide resultierenden Regler werden Simulationsergebnisse zu drei unterschiedlichen Szenarien betrachtet. Eine vergleichende Diskussion schließt diesen Abschnitt ab.

[52] Keine Aussage möglich, Lyapunov-Kandidat liefert ein hinreichendes Kriterium.

3.2.5.1 Lokale Solltrajektorienapproximation durch kubische Polynome

Der im vorliegenden Kapitel präsentierte Ansatz löst optimale Solltrajektorienfolgeregelungsprobleme, deren Gütefunktional ADP-kompatible Solltrajektorien berücksichtigt (vgl. (3.37)). Um den anwendungsnahen Fall beliebiger Sollvorgaben $\boldsymbol{x}_{\mathrm{r,soll},k}$ handhaben zu können, wird im Folgenden ein Verfahren vorgestellt, das aus einem beliebigen Sollzustandsverlauf, der auf einem endlichen, gleitenden Horizont der Länge n_{h} gegebenen ist, in jedem Zeitschritt k eine ADP-kompatible Approximation erzeugt. Exemplarisch[53] werden hierzu kubische Polynome gewählt, die einen Kompromiss zwischen der Approximationsfähigkeit in der lokalen Nachbarschaft des aktuellen Zeitschritts und einer moderaten Zunahme der Anzahl zu lernender Gewichte darstellen.

Damit $\boldsymbol{x}_{\mathrm{r}}\left(\boldsymbol{Z}_k^{(\kappa)}\right)$ kubischen Polynomen entspricht, werde

$$\boldsymbol{\rho}(\kappa) = \begin{bmatrix}(\kappa\Delta t)^3 & (\kappa\Delta t)^2 & \kappa\Delta t & 1\end{bmatrix}^{\mathsf{T}} \tag{3.67}$$

gewählt, wobei Δt die Abtastzeit darstellt[54]. Eine Transformationsmatrix $\boldsymbol{D}(\kappa)$, die zu einer verschobenen Parametermatrix $\boldsymbol{Z}_k^{(\kappa)}$ nach Definition 3.2 führt, folgt aus dem durch

$$\begin{aligned}
\boldsymbol{x}_{\mathrm{r}}\left(\boldsymbol{Z}_k^{(\kappa+j)}\right) &= \boldsymbol{x}_{\mathrm{r}}(\boldsymbol{Z}_k, \kappa + j) \\
&= \boldsymbol{Z}_k \boldsymbol{\rho}(\kappa + j) \\
&= \boldsymbol{Z}_k \begin{bmatrix}((\kappa+j)\Delta t)^3 \\ ((\kappa+j)\Delta t)^2 \\ (\kappa+j)\Delta t \\ 1\end{bmatrix} \\
&= \boldsymbol{Z}_k \underbrace{\begin{bmatrix}1 & 3\kappa\Delta t & 3(\kappa\Delta t)^2 & (\kappa\Delta t)^3 \\ 0 & 1 & 2\kappa\Delta t & (\kappa\Delta t)^2 \\ 0 & 0 & 1 & \kappa\Delta t \\ 0 & 0 & 0 & 1\end{bmatrix}}_{=:\boldsymbol{D}(\kappa)} \boldsymbol{\rho}(j) \\
&= \boldsymbol{Z}_k^{(\kappa)} \boldsymbol{\rho}(j)
\end{aligned} \tag{3.68}$$

[53] Die Approximationstheorie stellt ein eigenes Forschungsgebiet dar und soll in der vorliegenden Arbeit nicht vertieft werden. Für Untersuchungen hinsichtlich der Approximationsfähigkeit von Polynomen auf einem endlichen Intervall sei z. B. auf [DS10] verwiesen. Während nachfolgend beispielhaft kubische Polynome verwendet werden, werden im Anwendungskapitel (Kapitel 6) lineare (Abschnitt 6.1) und quadratische (Abschnitt 6.2) Polynome mit konstanten Sollzustandsvorgaben verglichen. Eine geeignete Wahl der ADP-kompatiblen Solltrajektorienapproximation, wie der zugehörige Polynomgrad oder der gleitende Vorausschauhorizont n_{h}, hängt insbesondere von der konkreten Anwendung ab. Somit können für jede konkrete Problemstellung beispielsweise der gewünschte Solltrajektorienverlauf oder auch mögliches Vorwissen über die Zeitkonstante des zu regelnden Systems bei der Wahl eines ADP-kompatiblen Solltrajektorienapproximators berücksichtigt werden.

[54] Andere Approximationen können durch eine andere Wahl von $\boldsymbol{\rho}(\kappa)$ erzielt werden. Beispielsweise führt $\boldsymbol{\rho}(\kappa) = \begin{bmatrix}\kappa\Delta t & 1\end{bmatrix}^{\mathsf{T}}$ zu linearer Interpolation und $\boldsymbol{\rho}(\kappa) = 1$ zu einer konstanten Sollzustandsvorgabe.

gegebenen Zusammenhang[55]. Für einen beliebigen Solltrajektorienverlauf $\boldsymbol{x}_{\mathrm{r,soll},k}$ wird ein Parameter $\boldsymbol{Z}_k$ in jedem Zeitschritt k benötigt, sodass

$$\boldsymbol{x}_{\mathrm{r}}\left(\boldsymbol{Z}_k^{(\kappa)}\right) = \begin{bmatrix} x_{\mathrm{r},1}\left(\boldsymbol{Z}_k^{(\kappa)}\right) \\ \vdots \\ x_{\mathrm{r},n}\left(\boldsymbol{Z}_k^{(\kappa)}\right) \end{bmatrix}, \quad \kappa = 0, 1, \dots, \tag{3.69}$$

eine Approximation von

$$\boldsymbol{x}_{\mathrm{r,soll},k+\kappa} = \begin{bmatrix} x_{\mathrm{r,soll},k+\kappa,1} \\ \vdots \\ x_{\mathrm{r,soll},k+\kappa,n} \end{bmatrix} \tag{3.70}$$

darstellt. Um diesen Parameter $\boldsymbol{Z}_k$ zu bestimmen, der den lokalen Solltrajektorienverlauf approximiert, sei die beliebige Solltrajektorie zur Laufzeit über einen gleitenden Vorausschauhorizont von n_{h} Zeitschritten bekannt, d. h. $\boldsymbol{x}_{\mathrm{r,soll},k:k+n_{\mathrm{h}}-1}$ sei gegeben. Dann kann $\boldsymbol{Z}_k$ beispielsweise mithilfe einer gewichteten Least-Squares-Regression bestimmt werden. Sei hierzu

$$\begin{aligned} \boldsymbol{x}_{\mathrm{r,soll},k:k+n_{\mathrm{h}}-1} &:= \begin{bmatrix} \boldsymbol{x}_{\mathrm{r,soll},k} & \cdots & \boldsymbol{x}_{\mathrm{r,soll},k+n_{\mathrm{h}}-1} \end{bmatrix}, \\ \boldsymbol{\rho}_{0:n_{\mathrm{h}}-1} &:= \begin{bmatrix} \boldsymbol{\rho}^{\mathsf{T}}(0) \\ \boldsymbol{\rho}^{\mathsf{T}}(1) \\ \dots \\ \boldsymbol{\rho}^{\mathsf{T}}(n_{\mathrm{h}}-1) \end{bmatrix} \text{ und } \boldsymbol{W}_{\mathrm{p}} := \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \beta & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \beta^{n_{\mathrm{h}}-1} \end{bmatrix}, \end{aligned} \tag{3.71}$$

wobei $\boldsymbol{W}_{\mathrm{p}}$ die Gewichtungsmatrix mit den Least-Squares-Gewichtungen β darstellt. Nachfolgend werde beispielhaft $\beta = \gamma$ gewählt, sodass analog zur Diskontierung der Kosten (vgl. J_k in (3.16) und (3.37)) zukünftige Zeitschritte weniger wichtig bei der Least-Squares-Regression sind. Folglich ergibt sich die lokale Solltrajektorienapproximation durch den Parameter

$$\boldsymbol{Z}_k = \boldsymbol{x}_{\mathrm{r,soll},k:k+n_{\mathrm{h}}-1} \underbrace{\boldsymbol{W}_{\mathrm{p}} \boldsymbol{\rho}_{0:n_{\mathrm{h}}-1} \left(\boldsymbol{\rho}_{0:n_{\mathrm{h}}-1}^{\mathsf{T}} \boldsymbol{W}_{\mathrm{p}} \boldsymbol{\rho}_{0:n_{\mathrm{h}}-1}\right)^{-1}}_{=:\boldsymbol{P}_{\mathrm{LS}}}. \tag{3.72}$$

Aus Anwendungssicht besonders günstig ist hierbei die Tatsache, dass sich bei gegebenem n_{h} und gegebenem $\boldsymbol{\rho}(\cdot)$ die Matrix $\boldsymbol{P}_{\mathrm{LS}}$ im Voraus berechnen lässt, sodass sich die Berechnung von $\boldsymbol{Z}_k$ nach (3.72) auf eine einzelne Matrixmultiplikation beschränkt. Für den exemplarisch gewählten gleitenden Vorausschauhorizont $n_{\mathrm{h}} = 10$, der auch im folgenden Simulationsbeispiel verwendet wird, ist ein beispielhafter Ausschnitt eines Solltrajektorienverlaufs $x_{\mathrm{r,soll},k,1}$ des ersten Zustands x_1 sowie dessen lokale Approximation für die Beispielzeitschritte $k = 105$ und $k = 106$ in Abbildung 3.2 veranschaulicht.

[55] Mittels Koeffizientenvergleich ist zudem ersichtlich, dass für Polynome vom Grad d allgemein
$$\boldsymbol{D}(\kappa) = \begin{bmatrix} \binom{d}{0}(\kappa\Delta t)^0 & \binom{d}{1}(\kappa\Delta t)^1 & \dots & \binom{d}{d}(\kappa\Delta t)^d \\ 0 & \binom{d-1}{0}(\kappa\Delta t)^0 & \dots & \binom{d-1}{d-1}(\kappa\Delta t)^{d-1} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \binom{0}{0}(\kappa\Delta t)^0 \end{bmatrix}$$ gewählt werden kann.

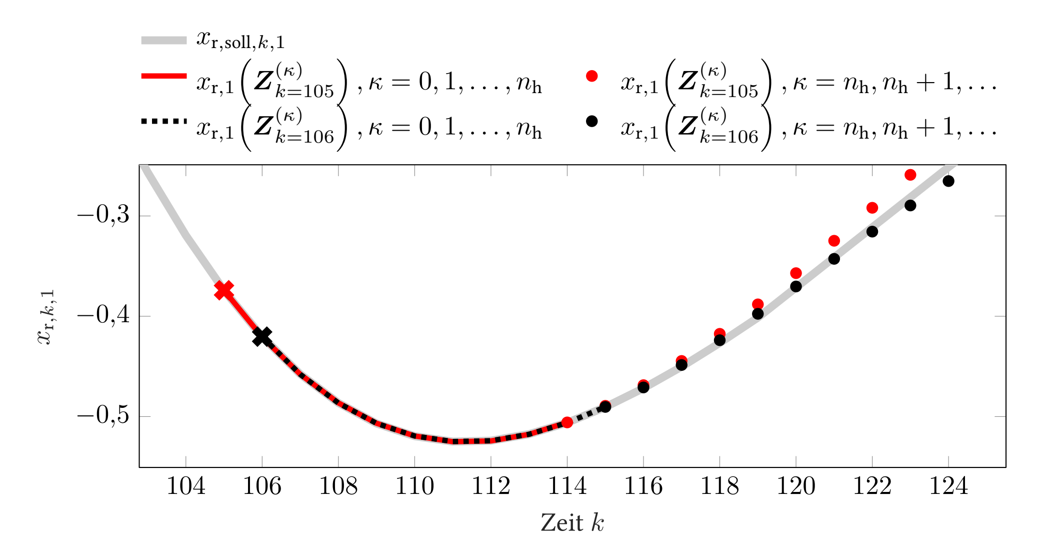


Abbildung 3.2: Beispielhafter Ausschnitt des Solltrajektorienverlaufs $x_{\mathrm{r,soll},k,1}$ des ersten Zustands x_1 (grau) sowie die lokale Approximation durch kubische Polynome zu den Zeitschritten $k = 105$ (rot) und $k = 106$ (schwarz), die für $\boldsymbol{Z}_{k=105}$ bzw. $\boldsymbol{Z}_{k=106}$ resultieren. Der Vorausschauhorizont, der für die lokale Solltrajektorienapproximation, d. h. die Anpassung des Solltrajektorienparameters $\boldsymbol{Z}_k$ in jedem Zeitschritt k, genutzt wird, ist in diesem Beispiel $n_{\mathrm{h}} = 10$ (rote durchgezogene und schwarze gestrichelte Linie).

3.2.5.2 Beispielsystem und Gütefunktional

Im Folgenden werde ein Feder-Masse-Dämpfer-System mit der Systemdynamik

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 0 & 1 \\ -\frac{c_{\mathrm{sys}}}{m_{\mathrm{sys}}} & -\frac{d_{\mathrm{sys}}}{m_{\mathrm{sys}}} \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0 \\ \frac{1}{m_{\mathrm{sys}}} \end{bmatrix} u(t) \tag{3.73}$$

mit $m_{\mathrm{sys}} = 0{,}5\,\mathrm{kg}$, $c_{\mathrm{sys}} = 0{,}1\,\mathrm{N\,m^{-1}}$ und $d_{\mathrm{sys}} = 0{,}1\,\mathrm{kg\,s^{-1}}$ betrachtet. Die Diskretisierung dieses Systems mittels Tustin-Approximation mit einer Abtastzeit von $\Delta t = 0{,}1\,\mathrm{s}$ liefert

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} 0{,}9990 & 0{,}0990 \\ -0{,}0198 & 0{,}9792 \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} 0{,}0099 \\ 0{,}1979 \end{bmatrix} u_k. \tag{3.74}$$

Durch x_1 wird die Position und durch x_2 die Geschwindigkeit der Masse m_{sys} beschrieben, während die Stellgröße u_k einer auf die Masse wirkenden Kraft entspricht. Die durch (3.73) bzw. (3.74) beschriebene Systemdynamik sei dem Regler im Folgenden nicht bekannt.

Mit dem Ziel, die Position der Masse (d. h. x_1) einem Solltrajektorienverlauf folgen zu lassen, werde

$$\boldsymbol{Q} = \begin{bmatrix} 100 & 0 \\ 0 & 0 \end{bmatrix} \text{ und } \boldsymbol{R} = 1 \tag{3.75}$$

gesetzt, um Abweichungen des ersten Systemzustands vom parametrierten Sollzustandsverlauf nach (3.37) zu bestrafen. Der Diskontierungsfaktor werde zu $\gamma = 0{,}9$ gewählt. Für dieses Beispiel sind Annahme 3.1 und Annahme 3.2 erfüllt, d. h. nach Satz 3.1 existiert der optimale Regler $\boldsymbol{\mu}^*\left(\boldsymbol{x}_{k+\kappa}, \boldsymbol{Z}_k^{(\kappa)}\right)$ und ist zudem eindeutig. Außerdem folgt aus Satz 3.3 $b_\gamma = 0{,}6263$. Somit führt die optimale Lösung des mit $\gamma = 0{,}9 > b_\gamma$ diskontierten Optimierungsproblems zu einem nach Definition 3.5 Referenz-Zustands-stabilen System.

3.2.5.3 Trainingsvorgang

Der in diesem Kapitel vorgestellte ADP-basierte Solltrajektorienfolgeregler, der mit einer ADP-kompatiblen, parametrischen Darstellung des Sollzustandsverlaufs arbeitet, wird im Folgenden trainiert, um anschließend validiert zu werden. Zu Vergleichszwecken wird zudem ein ADP-Trajektorienfolgeregelungsansatz verwendet, dem, genau wie in den Arbeiten von Luo et al. [LLHW16] und Kiumarsi et al. [KLM$^+$14], die Annahme zugrunde liegt, der Solltrajektorienverlauf werde direkt und global durch ein Exosystem $\boldsymbol{x}_{\mathrm{r},k+1} = \boldsymbol{f}_{\boldsymbol{x}_\mathrm{r}}(\boldsymbol{x}_{\mathrm{r},k})$ erzeugt (vgl. auch Abschnitt 2.2.2). Somit hängt die Q-Function $Q(\boldsymbol{x}_k, \boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{u}_k)$ (vgl. [KLM$^+$14]) nur vom aktuellen Sollzustand $\boldsymbol{x}_{\mathrm{r},k}$ im Zeitschritt k ab, wohingegen der Parameter $\boldsymbol{Z}_k$ der vorgestellten PRADP-Methode potenziell mehr Information über den Sollzustandsverlauf beinhalten kann. Für die Vergleichsmethode ist nach [KLM$^+$14, Abschnitt 5.1] die Q-Function $Q(\boldsymbol{x}_k, \boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{u}_k)$ quadratisch bezüglich $\boldsymbol{x}_k$, $\boldsymbol{u}_k$ und $\boldsymbol{x}_{\mathrm{r},k}$, sofern die Exosystemdynamik $\boldsymbol{f}_{\boldsymbol{x}_\mathrm{r}}(\boldsymbol{x}_{\mathrm{r},k})$ linear ist, und die Policy Iteration nach [KLM$^+$14, Algorithm 3] kann durchgeführt werden.

Die beiden untersuchten modellfreien ADP-Trajektorienfolgeregelungsmethoden werden jeweils mit $M = 500$ Datentupeln $\mathcal{T}_k$ trainiert. Während des Datenaufzeichnens wird zur Systemanregung Rauschen, das aus einer Normalverteilung mit Mittelwert 0 und Standardabweichung 1 erzeugt wird, zur Stellgröße u_k addiert. Der Solltrajektorienverlauf während des Trainings werde durch

$$\begin{bmatrix} x_{\mathrm{r,soll},k+1,1} \\ x_{\mathrm{r,soll},k+1,2} \end{bmatrix} = \boldsymbol{x}_{\mathrm{r,soll},k+1} = \boldsymbol{f}_{\boldsymbol{x}_\mathrm{r}}(\boldsymbol{x}_{\mathrm{r,soll},k}) = \underbrace{\begin{bmatrix} 0{,}9988 & 0{,}0500 \\ -0{,}0500 & 0{,}9988 \end{bmatrix}}_{\boldsymbol{F}_\mathrm{ref}} \boldsymbol{x}_{\mathrm{r,soll},k} \tag{3.76}$$

mit dem Initialzustand $\boldsymbol{x}_{\mathrm{r,soll},0} = \begin{bmatrix} 0 & 1 \end{bmatrix}^\mathsf{T}$ beschrieben[56]. Die Vergleichsmethode gemäß [KLM$^+$14] bzw. [LLHW16] verwendet $\boldsymbol{x}_{\mathrm{r},k} = \boldsymbol{x}_{\mathrm{r,soll},k}$, wohingegen für die PRADP-Methode der nach Abschnitt 3.2.5.1 bestimmte Parameter $\boldsymbol{Z}_k$, der ein kubisches Polynom parametriert und den Sollzustandsverlauf ab dem jeweiligen Zeitschritt k beschreibt, genutzt wird.

Der PRADP-basierte Regler wird dann entsprechend Algorithmus 3.1 trainiert, wobei die Abbruchbedingung $e_{\hat{\boldsymbol{w}}} = 10^{-6}$ gewählt wird. Der Policy-Improvement-Schritt erfolgt nach (3.66). Aufgrund von Lemma 3.2 werden Basisfunktionen $\boldsymbol{\phi}(\cdot)$ gewählt, die quadratisch bezüglich $\boldsymbol{x}_k$, $\boldsymbol{u}_k$ und $\boldsymbol{z}_{k,1:n}$ sind. Weiterhin wird mangels besseren Wissens $\hat{\boldsymbol{w}}^{[0]}$ so initialisiert, dass

56 Diese lineare Exosystemdynamik wird insbesondere verwendet, um die Voraussetzungen der Vergleichsmethode zu erfüllen. Für die PRADP-Methode wäre dies, wie auch im Anwendungskapitel 6 ersichtlich, grundsätzlich nicht nötig, da die Verwendung von $\boldsymbol{Z}_k$ und $\boldsymbol{Z}_k^{(1)}$ in (3.26) ADP-Kompatibilität gewährleistet.

$\hat{\boldsymbol{\mu}}^{[0]} = \mathbf{0}$ gilt[57]. Der gesamte Ablauf ist in Abbildung 3.3 veranschaulicht. Die Vergleichsmethode ist wie in [KLM+14, Algorithm 3] beschrieben implementiert, wobei ebenfalls die Abbruchbedingung $e_{\hat{\boldsymbol{w}}} = 10^{-6}$ sowie $\hat{\boldsymbol{\mu}}^{[0]} = \mathbf{0}$ genutzt wird.

3.2.5.4 Untersuchung der gelernten Regler

Zunächst werden die aus Daten und ohne Kenntnis der Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ gelernten ADP-Regler mit den analytisch berechneten Grundwahrheiten verglichen. Anschließend werden verschiedene Simulationsszenarien betrachtet, um Unterschiede zwischen dem PRADP-Regler und dem Vergleichsverfahren hinsichtlich ihrer Flexibilität und Performanz herauszuarbeiten.

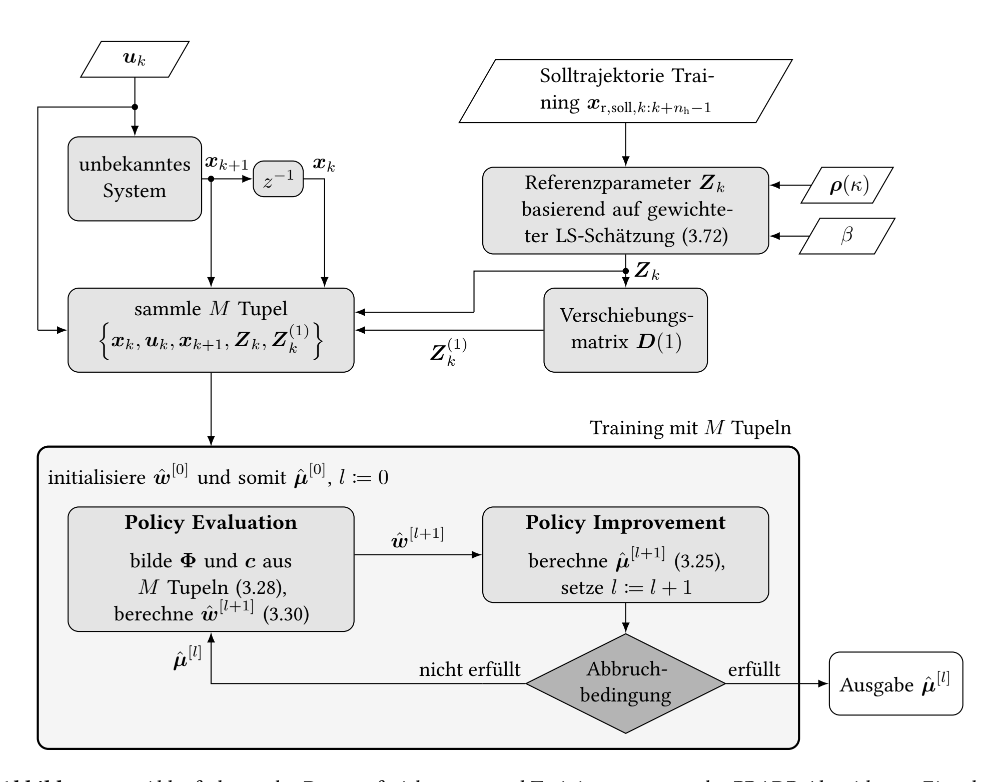


Abbildung 3.3: Ablaufschema des Datenaufzeichnungs- und Trainingsprozesses des PRADP-Algorithmus. Eingabegrößen sind hierbei die Stellgröße $\boldsymbol{u}_k$ und Solltrajektorie $\boldsymbol{x}_{\mathrm{r,soll},k:k+n_\mathrm{h}-1}$ während des Trainings, die Basisfunktionen $\boldsymbol{\rho}(\kappa)$ zur Referenzapproximation und der Parameter β für die gewichtete Least-Squares-Approximation. Sobald die Abbruchbedingung (siehe Algorithmus 3.1) erfüllt ist, wird der geschätzte optimale Regler $\hat{\boldsymbol{\mu}}^{[l]}$ ausgegeben.

[57] Dies kann erreicht werden, indem die Gewichte, die zu $\boldsymbol{h}_{\mathrm{uu}}$ gehören, so gesetzt werden, dass $\boldsymbol{h}_{\mathrm{uu}} \succ \mathbf{0}$ gilt, wohingegen alle anderen Gewichte zu null gesetzt werden (vgl. Lemma 3.2).

Für die Verwendung der parametrierten Solltrajektorie ergibt sich die optimale Reglermatrix $\boldsymbol{K}^*$, die unter Verwendung des vollständigen Systemwissens nach Satz 3.1 und Bemerkung 3.5 berechnet werden kann, zu

$$\boldsymbol{\mu}^*(\boldsymbol{x}_k, \boldsymbol{Z}_k) = -\underbrace{\begin{bmatrix} 6{,}30 & 2{,}26 & -0{,}31 & -0{,}97 & -2{,}37 & -6{,}40 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\boldsymbol{K}^*} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{z}_{k,1:n} \end{bmatrix}. \tag{3.77}$$

Der Vergleich des gelernten PRADP-Reglers $\boldsymbol{K}_{\text{PRADP}}$ mit dieser Grundwahrheit $\boldsymbol{K}^*$ liefert $\|\boldsymbol{K}_{\text{PRADP}} - \boldsymbol{K}^*\|_2 = 6{,}51 \cdot 10^{-14}$. Somit stimmt der ohne Kenntnis der Systemdynamik gelernte PRADP-Regler bis auf numerische Ungenauigkeiten mit der optimalen Lösung überein.

Für die Vergleichsmethode ergibt sich der optimale Regler, nachfolgend durch $\boldsymbol{\mu}^*_{\text{exo}}(\cdot)$ und die Reglermatrix $\boldsymbol{K}^*_{\text{exo}}$ bezeichnet, für das gegebene Beispielproblem nach [KLM+14, (58)] zu

$$\boldsymbol{\mu}^*_{\text{exo}}(\boldsymbol{x}_k, \boldsymbol{x}_{\text{r},k}) = -\underbrace{\begin{bmatrix} 6{,}30 & 2{,}26 & -6{,}28 & -1{,}18 \end{bmatrix}}_{\boldsymbol{K}^*_{\text{exo}}} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{x}_{\text{r},k} \end{bmatrix}. \tag{3.78}$$

Die Reglermatrix $\boldsymbol{K}_{\text{exo, ADP}}$, die wie in Abschnitt 3.2.5.3 beschrieben aus Daten gelernt wird, stimmt ebenfalls mit der Grundwahrheit überein, da $\|\boldsymbol{K}_{\text{exo, ADP}} - \boldsymbol{K}^*_{\text{exo}}\|_2 = 2{,}60 \cdot 10^{-13}$ gilt. Somit liegt auch hier ein erfolgreich gelernter Regler vor.

Um die Performanz des in dieser Arbeit vorgestellten PRADP-Reglers und der Vergleichsmethode zu untersuchen, werden drei unterschiedliche Szenarien betrachtet. In allen Szenarien werden die zuvor gelernten Regler ohne weitere Anpassungen verwendet, zudem wird jeweils der Anfangszustand zu $\boldsymbol{x}_0 = \begin{bmatrix} 1 & 0 \end{bmatrix}^\mathsf{T}$ gesetzt. Die unterschiedlichen Szenarien werden im Folgenden beschrieben.

1. Der Sollzustandsverlauf wird durch dieselbe Exosystemdynamik erzeugt, die während des Trainings verwendet wurde, d. h. $\boldsymbol{F}_{\text{ref}}$ wie in (3.76). Der resultierende Verlauf des Zustands x_1 ist in Abbildung 3.4a gezeigt. In Abbildung 3.4b sind die Parameter $\boldsymbol{z}_{k,1} = \begin{bmatrix} z_{k,1,1} & \dots & z_{k,1,n_z} \end{bmatrix}^\mathsf{T}$ mit $n_z = 4$ gegeben. Diese parametrieren das kubische Polynom, das $x_{\text{r},1}(\boldsymbol{Z}_k, 0)$ beschreibt.

2. Der Sollzustandsverlauf wird ebenfalls durch eine zeitinvariante Exosystemdynamik erzeugt, jedoch durch eine andere Dynamik als die während des Trainings verwendete, d. h. es gilt $\boldsymbol{F}_{\text{ref, val}} \neq \boldsymbol{F}_{\text{ref}}$. Der Verlauf des Sollzustands wird durch

$$\boldsymbol{x}_{\text{r},k+1} = \underbrace{\begin{bmatrix} 0{,}9987 & 0{,}0030 \\ -0{,}1998 & 0{,}9987 \end{bmatrix}}_{=:\boldsymbol{F}_{\text{ref, val}}} \boldsymbol{x}_{\text{r},k} \tag{3.79}$$

mit $\boldsymbol{x}_{\text{r},0} = \begin{bmatrix} 10 & 1 \end{bmatrix}^\mathsf{T}$ beschrieben. Der resultierende Verlauf des Zustands x_1 sowie die Polynomkoeffizienten, die den Sollverlauf approximieren, sind in Abbildung 3.5 gegeben. Um darüber hinaus die Qualität der Trajektorienfolgeregler im Sinne der resultierenden Kosten vergleichen zu können, sind die Einschrittkosten $r(\boldsymbol{x}_k, \boldsymbol{x}_{\text{r}}(\boldsymbol{Z}_k), \boldsymbol{u}_k)$ sowie die akkumulierten Kosten $\sum_{i=0}^{k} r(\boldsymbol{x}_i, \boldsymbol{x}_{\text{r}}(\boldsymbol{Z}_i), \boldsymbol{u}_i)$ gezeigt.

3. Im dritten Szenario wird der Sollzustandsverlauf nicht durch ein zeitinvariantes Exosystem generiert, sondern eine benutzerdefinierte Vorgabe für $x_{\mathrm{r,soll},k,1}$ verwendet und anschließend, wie in Abschnitt 3.2.5.1 beschrieben, durch $x_{\mathrm{r},k,1}$ approximiert. Der sich ergebende Sollverlauf ist in Abbildung 3.6a gezeigt. Zudem wird in diesem Beispiel $x_{\mathrm{r},k,2} = 0, \forall k$, gesetzt. Dies ist dadurch motiviert, dass der zweite Systemzustand aufgrund der Wahl von $\boldsymbol{Q}$ in (3.75) ohnehin nicht bestraft wird. Die Ergebnisse sind in Abbildung 3.6 veranschaulicht.

3.2.5.5 Diskussion der Simulationsergebnisse

Wie den Abbildungen 3.4a, 3.5a und 3.6a zu entnehmen ist, kann der PRADP-Regler in allen drei Szenarien dem Sollzustandsverlauf folgen. Da der Regler durch den Parameter $\boldsymbol{Z}_k$ nicht nur Informationen über den im Zeitschritt k aktuellen Sollzustand, sondern auch über dessen (lokal approximierten) zukünftigen Verlauf besitzt, weist der PRADP-Regler vorausschauendes Verhalten auf, anstatt rein reaktiv zu handeln. Dies ist in den Abbildungen 3.4a, 3.5a und 3.6a zu sehen, da der Systemzustand dem Sollverlauf direkt folgt und es zu keiner zeitlichen Verzögerung kommt.

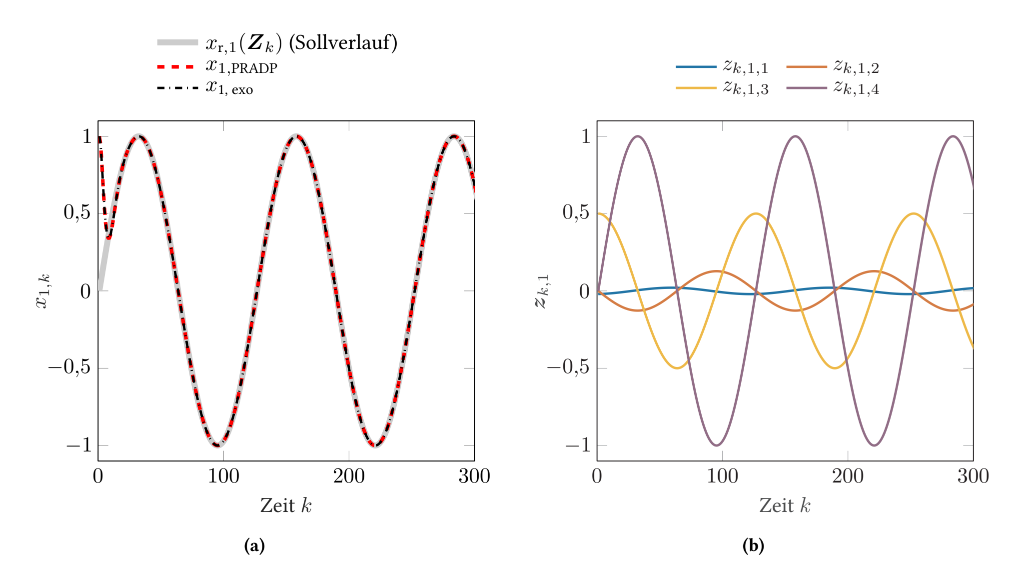


Abbildung 3.4: Ergebnis der vorgestellten PRADP-Methode im Vergleich zu einem Regler nach [LLHW16], [KLM+14] für Szenario 1, bei dem der Solltrajektorienverlauf durch $\boldsymbol{F}_{\mathrm{ref}}$ generiert wird.
(a): Der approximierte Solltrajektorienverlauf $x_{\mathrm{r},1}(\boldsymbol{Z}_k, 0)$ ist in Grau dargestellt, der resultierende Zustand bei Verwendung des vorgestellten PRADP-Algorithmus in Rot und das Ergebnis der Vergleichsmethode in Schwarz.
(b): Parametervektor $\boldsymbol{z}_{k,1}$ der kubischen Polynome, welche den Solltrajektorienverlauf $x_{\mathrm{r},1}(\boldsymbol{Z}_k, 0)$ beschreiben.

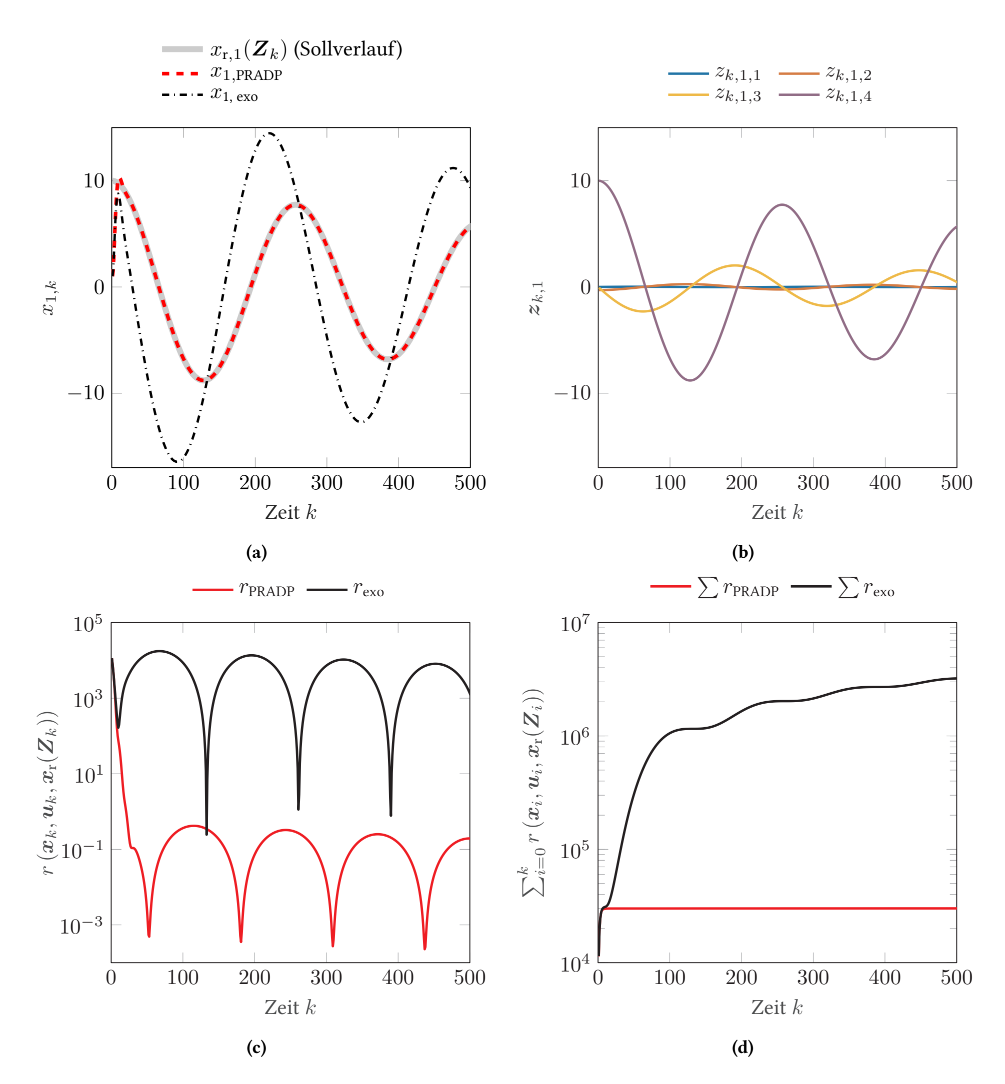


Abbildung 3.5: Ergebnis der vorgestellten PRADP-Methode im Vergleich zu einem Regler nach [LLHW16], [KLM+14] für Szenario 2, bei dem der Solltrajektorienverlauf durch $\boldsymbol{F}_{\mathrm{ref,\,val}} \neq \boldsymbol{F}_{\mathrm{ref}}$ generiert wird.
(a): Der approximierte Solltrajektorienverlauf $x_{\mathrm{r},1}(\boldsymbol{Z}_k, 0)$ ist in Grau dargestellt, der resultierende Zustand bei Verwendung des vorgestellten PRADP-Algorithmus in Rot und das Ergebnis der Vergleichsmethode in Schwarz.
(b): Parametervektor $\boldsymbol{z}_{k,1}$ der kubischen Polynome, welche den Solltrajektorienverlauf $x_{\mathrm{r},1}(\boldsymbol{Z}_k, 0)$ beschreiben.
(c): Die Einschrittkosten $r(\boldsymbol{x}_k, \boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_k), \boldsymbol{u}_k)$, welche bei Verwendung der PRADP-Methode entstehen, sind in Rot gezeigt, wohingegen die mit der Vergleichsmethode verbundenen Kosten in Schwarz visualisiert sind (logarithmische Ordinate).
(d): Die akkumulierten Kosten $\sum_{i=0}^{k} r(\boldsymbol{x}_i, \boldsymbol{x}_\mathrm{r}(\boldsymbol{Z}_i), \boldsymbol{u}_i)$ der PRADP-Methode sind in Rot, die der Vergleichsmethode in Schwarz gegeben (logarithmische Ordinate).

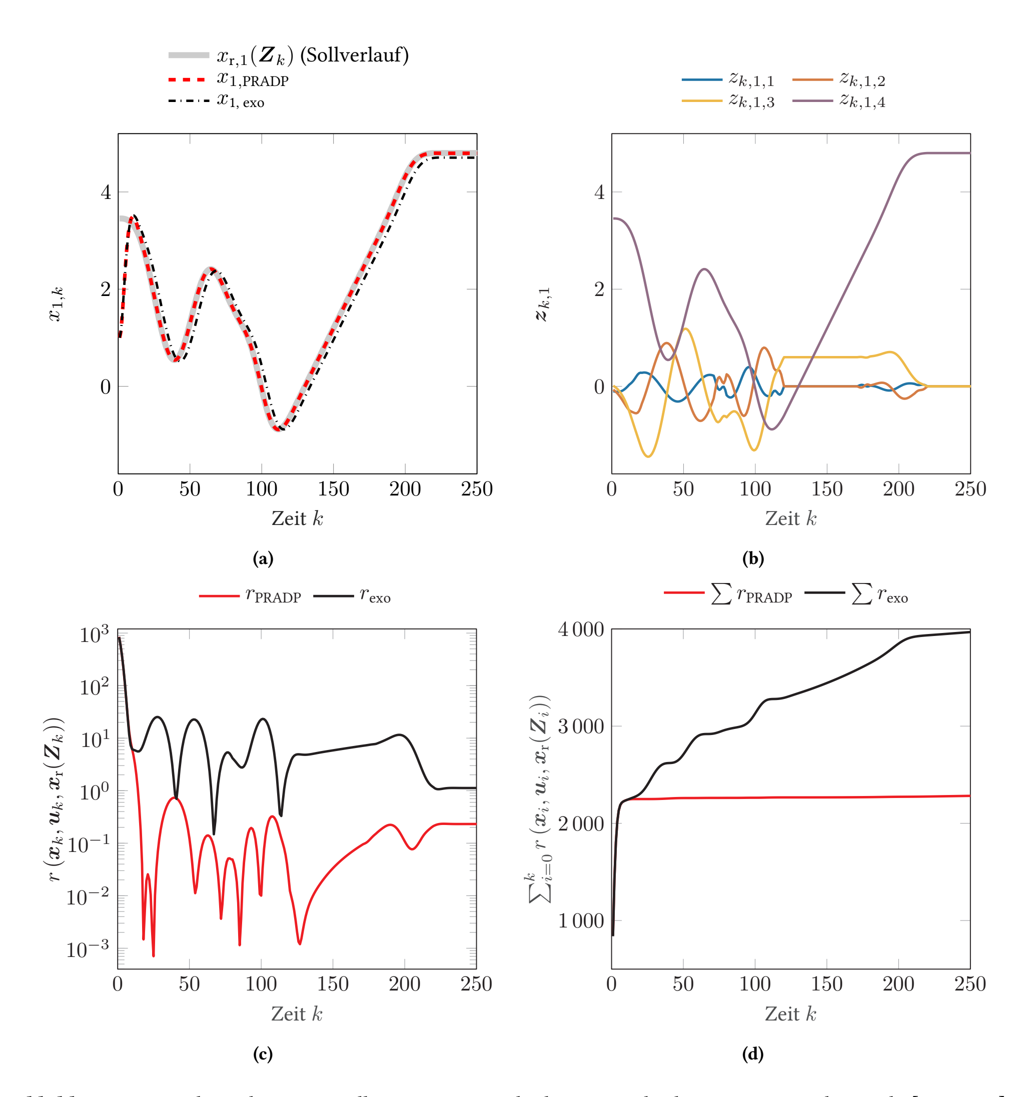


Abbildung 3.6: Ergebnis der vorgestellten PRADP-Methode im Vergleich zu einem Regler nach [LLHW16], [KLM+14] für Szenario 3, bei dem der Solltrajektorienverlauf keiner Dynamik folgt, sondern einer beliebigen benutzerdefinierten Vorgabe entspricht.
(a): Der approximierte Solltrajektorienverlauf $x_{r,1}(\boldsymbol{Z}_k, 0)$ ist in Grau dargestellt, der resultierende Zustand bei Verwendung des vorgestellten PRADP-Algorithmus in Rot und das Ergebnis der Vergleichsmethode in Schwarz.
(b): Parametervektor $\boldsymbol{z}_{k,1}$ der kubischen Polynome, welche den Solltrajektorienverlauf $x_{r,1}(\boldsymbol{Z}_k, 0)$ beschreiben.
(c): Die Einschrittkosten $r(\boldsymbol{x}_k, \boldsymbol{x}_r(\boldsymbol{Z}_k), \boldsymbol{u}_k)$, welche bei Verwendung der PRADP-Methode entstehen, sind in Rot gezeigt, wohingegen die mit der Vergleichsmethode verbundenen Kosten in Schwarz visualisiert sind (logarithmische Ordinate).
(d): Die akkumulierten Kosten $\sum_{i=0}^{k} r(\boldsymbol{x}_i, \boldsymbol{x}_r(\boldsymbol{Z}_i), \boldsymbol{u}_i)$ der PRADP-Methode sind in Rot, die der Vergleichsmethode in Schwarz gegeben.

Die Vergleichsmethode nach [LLHW16] bzw. [KLM+14] führt zu ähnlichen Ergebnissen, solange der Solltrajektorienverlauf mit der Exosystemdynamik $\boldsymbol{F}_{\mathrm{ref}}$, die während des Trainings verwendet wurde, übereinstimmt, wie dies in Szenario 1 der Fall ist (vgl. Abbildung 3.4). Dieser Vergleichsregler weist jedoch teilweise signifikante Abweichungen vom Solltrajektorienverlauf auf, sobald dieser nicht mehr durch die Dynamik $\boldsymbol{F}_{\mathrm{ref}}$ beschrieben wird, d. h. sobald (3.76) nicht mehr gilt. Dies offenbart sich in Szenario 2 (Abbildung 3.5) und Szenario 3 (Abbildung 3.6). Obwohl der Vergleichsregler dem Sollzustand in Szenario 3 näherungsweise folgt[58], hat der Regler keinerlei Kenntnis über den weiteren Verlauf des Sollzustands. Durch diese mangelnde Vorausschaufähigkeit kann der Regler nur reagieren und es kommt zu einem zeitlichen Versatz zwischen der Trajektorie des Zustands x_1 und dem gewünschten Verlauf.

Zudem offenbaren die Einschrittkosten in den Abbildungen 3.5c und 3.6c sowie die akkumulierten Kosten in den Abbildungen 3.5d und 3.6d, dass die Kosten der Vergleichsmethode signifikant über den Kosten des PRADP-Ansatzes liegen, sobald der Solltrajektorienverlauf nicht mehr durch die während des Trainings verwendete Exosystemdynamik $\boldsymbol{F}_{\mathrm{ref}}$ beschrieben wird. Dies ist insbesondere auf größere Abweichungen von der Solltrajektorie zurückzuführen.

Abschließend lässt sich zusammenfassen, dass die beiden betrachteten ADP-Methoden zwar in Szenario 1 vergleichbare Ergebnisse liefern, die in dieser Arbeit vorgestellte PRADP-Methode jedoch in Szenario 2 und Szenario 3 überlegen ist. Dies liegt insbesondere daran, dass der PRADP-Regler zur Laufzeit stets lokal den aktuellen Solltrajektorienverlauf mithilfe des Parameters $\boldsymbol{Z}_k$ approximiert und daher im Gegensatz zum Vergleichsregler nicht annehmen muss, dass der Sollzustandsverlauf derselben Exosystemdynamik wie während des Trainingsvorgangs folgt.

3.3 Zeitdiskrete ADP-kompatible Referenztrajektorie auf einem endlichem Vorausschauhorizont

Nachdem im letzten Abschnitt eine ADP-kompatible parametrierte Darstellung des Sollzustandsverlaufs vorgestellt und in eine Q-Function integriert wurde, wird im vorliegenden Abschnitt die Verwendung eines beliebigen Sollzustandsverlaufs auf einem endlichen Vorausschauhorizont untersucht[59]. Hierzu wird zunächst eine ADP-kompatible Problemformulierung gegeben. Konkret wird ein linear-quadratisches[60] Optimierungsproblem mit unendlichem Zeithorizont angesetzt (vgl. Bemerkung 2.1). In dieses Optimierungsproblem geht in jedem Zeitschritt k der beliebige, auf einem gleitenden Vorausschauhorizont der Länge n_{h} gegebene, Sollzustandsverlauf ein. Um eine ADP-kompatible Darstellung nach Definition 3.1 zu erhalten und das Kostenfunktional auf einem unendlichen Zeithorizont definieren zu können, wird

[58] Dies ist dem Zustandsrückführungsterm $\boldsymbol{K}^{*}_{\mathrm{x,exo}}$ des Reglers $\boldsymbol{K}^{*}_{\mathrm{exo}}$, der identisch zum Zustandsrückführungsterm des PRADP-Reglers ist, sowie dem Regleranteil, welcher der aktuellen Sollzustandsposition zugeordnet ist und der bei beiden Methoden ähnliche Werte aufweist (d. h. $-6{,}40$ in (3.77) bzw. $-6{,}28$ in (3.78)), zu verdanken.

[59] Teile des vorliegenden Abschnitts wurden in einer wissenschaftlichen Fachzeitschrift veröffentlicht [KWFH20].

[60] Linear-quadratische Optimierungsprobleme sind für zahlreiche regelungstechnische Probleme relevant.

der Sollzustandsverlauf jenseits des Vorausschauhorizonts als konstant angenommen. Für diese Problemstellung lässt sich eine Q-Function definieren, die den beliebigen Sollverlauf auf dem gleitenden Horizont explizit berücksichtigt. Eine genaue Analyse der Form dieser erweiterten Q-Function für linear-quadratische Problemstellungen führt einerseits zu Existenz- und Eindeutigkeitsaussagen bezüglich der optimalen Lösung und erlaubt schließlich eine kompakte Approximationsstruktur. Diese Struktur wird anschließend genutzt, um mithilfe einer modellfreien ADP-Methode das optimale Regelgesetz zu erlernen. Konvergenzaussagen der iterativen Off-policy-Methode werden auf den hier vorgestellten Solltrajektorienfolgeregelungsfall übertragen. Simulationsergebnisse, die den prädiktiven Charakter der vorgestellten Methode offenbaren, und eine anschließende Diskussion runden diesen Abschnitt ab.

3.3.1 Problemdefinition

Betrachtet werde ein lineares, zeitdiskretes System

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \tag{3.80}$$

mit den diskreten Zeitschritten $k \in \mathbb{N}_{\geq 0}$, dem Zustandsvektor $\boldsymbol{x}_k \in \mathbb{R}^n$ und dem Eingangsvektor $\boldsymbol{u}_k \in \mathbb{R}^p$. Es wird angenommen, dass das System $(\boldsymbol{A}, \boldsymbol{B})$ steuerbar ist, jedoch seien $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ und $\boldsymbol{B} \in \mathbb{R}^{n \times p}$ unbekannt.

In jedem Zeitschritt k sei der beliebige Sollzustandsverlauf $\boldsymbol{x}^{\{k\}}_{\mathrm{r,soll},i} \in \mathbb{R}^n$ auf einem Vorausschauhorizont mit einer Länge von n_h Zeitschritten gegeben, wobei i den Zeitindex auf diesem Vorausschauhorizont bezeichnet. Jenseits des Horizonts n_h werde der Sollzustandsverlauf zu $\boldsymbol{0}$ angenommen. Somit ergibt sich aus Sicht des Zeitschritts k insgesamt der Sollzustandsverlauf

$$\boldsymbol{x}^{\{k\}}_{\mathrm{r},i} := \begin{cases} \boldsymbol{x}^{\{k\}}_{\mathrm{r,soll},i}, & \text{für } i = k, \dots, k + n_\mathrm{h} \\ \boldsymbol{0}, & \text{für } i > k + n_\mathrm{h}. \end{cases} \tag{3.81}$$

Das Ziel ist im Folgenden, ohne Kenntnis von $\boldsymbol{A}$ und $\boldsymbol{B}$ ein Regelgesetz zu bestimmen, sodass der Systemzustand $\boldsymbol{x}_i$ dem Solltrajektorienverlauf $\boldsymbol{x}^{\{k\}}_{\mathrm{r},i}$ (3.81) optimal bezüglich des Gütefunktionals

$$\begin{aligned} J_k &= \sum_{i=k}^{\infty} \gamma^{i-k} \frac{1}{2} \left((\boldsymbol{x}_i - \boldsymbol{x}_{\mathrm{r},i})^\mathsf{T} \boldsymbol{Q} \left(\boldsymbol{x}_i - \boldsymbol{x}_{\mathrm{r},i} \right) + \boldsymbol{u}_i^\mathsf{T} \boldsymbol{R} \boldsymbol{u}_i \right) \\ &=: \sum_{i=k}^{\infty} \gamma^{i-k} \underbrace{\frac{1}{2} \left(\boldsymbol{e}_i^\mathsf{T} \boldsymbol{Q} \boldsymbol{e}_i + \boldsymbol{u}_i^\mathsf{T} \boldsymbol{R} \boldsymbol{u}_i \right)}_{=: r(\boldsymbol{x}_i, \boldsymbol{u}_i, \boldsymbol{x}_{\mathrm{r},i}) =: r_i}, \end{aligned} \tag{3.82}$$

welches es zu minimieren gilt, folgt. Hierbei wird die Kurzschreibweise $\boldsymbol{x}_{\mathrm{r},i} := \boldsymbol{x}^{\{k\}}_{\mathrm{r},i}$ verwendet. Zudem bezeichnet $\boldsymbol{e}_i := \boldsymbol{x}_i - \boldsymbol{x}_{\mathrm{r},i}$ im Zeitschritt i die Abweichung des Systemzustands $\boldsymbol{x}_i$ vom Sollzustand $\boldsymbol{x}_{\mathrm{r},i}$. Weiterhin sei $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$, $\boldsymbol{Q} = \boldsymbol{Q}^\mathsf{T} \succeq \boldsymbol{0}$, eine symmetrische, positiv semidefinite Matrix, welche Abweichungen des Zustands $\boldsymbol{x}_i$ vom Sollzustand $\boldsymbol{x}_{\mathrm{r},i}$

bestraft. Zudem sei $(\boldsymbol{A}, \boldsymbol{C})$ detektierbar, wobei $\boldsymbol{C}$ so definiert ist, dass $\boldsymbol{C}^\mathsf{T}\boldsymbol{C} = \boldsymbol{Q}$ gilt. Der Stellaufwand wird durch die symmetrische, positiv definite Matrix $\boldsymbol{R} \in \mathbb{R}^{m\times m}$, $\boldsymbol{R} = \boldsymbol{R}^\mathsf{T} \succ \boldsymbol{0}$, bestraft. Durch $r_i \in \mathbb{R}$ werden die Einschrittkosten bezeichnet und $\gamma \in (0, 1]$ stellt einen Diskontierungsfaktor dar[61].

Die Wahl des Kostenfunktionals (3.82) ist dabei wie folgt motiviert: Einerseits ist der Sollzustandsverlauf üblicherweise nur begrenzt in die Zukunft bekannt (beispielsweise der Straßenverlauf beim autonomen Fahren), daher wird ein endlicher Vorausschauhorizont n_h verwendet. Andererseits basiert die in Abschnitt 3.3.3 verwendete ADP-Methode auf einem Kostenfunktional mit unendlichem Zeithorizont, wodurch eine effiziente Darstellung der Bellman-Gleichung ermöglicht wird (vgl. Bemerkung 2.1). Daher wird das Kostenfunktional (3.82) auf einem unendlichen Optimierungshorizont definiert, berücksichtigt dabei jedoch einen auf einem endlichen Vorausschauhorizont gegebenen Sollzustandsverlauf. Die nachfolgende Proposition zeigt, dass für die gegebene Problemstellung ein eindeutiger optimaler linearer Regler existiert.

Annahme 3.3
Sei $(\boldsymbol{A}, \boldsymbol{B})$ steuerbar, $\boldsymbol{Q} = \boldsymbol{Q}^\mathsf{T} \succeq \boldsymbol{0}$ und $\boldsymbol{R} = \boldsymbol{R}^\mathsf{T} \succ \boldsymbol{0}$, sowie $(\boldsymbol{A}, \boldsymbol{C})$ detektierbar mit $\boldsymbol{C}^\mathsf{T}\boldsymbol{C} = \boldsymbol{Q}$.

Proposition 3.3
Unter Annahme 3.3 existiert ein eindeutiger optimaler linearer Regler $\boldsymbol{\mu}^*(\tilde{\boldsymbol{x}}_k)$ mit $\tilde{\boldsymbol{x}}_k := \tilde{\boldsymbol{x}}_{k,k} := \begin{bmatrix}\boldsymbol{x}_k^\mathsf{T} & \boldsymbol{x}_{\mathrm{r},k}^{\{k\}\mathsf{T}} & \dots & \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}}^{\{k\}\mathsf{T}}\end{bmatrix}^\mathsf{T}$, der das durch (3.80)–(3.82) beschriebene optimale Solltrajektorienfolgeregelungsproblem löst.

Beweis:
Die zeitliche Propagation des Sollzustands $\boldsymbol{x}_{\mathrm{r},i}^{\{k\}}$ in (3.81) kann durch

$$\begin{bmatrix}\boldsymbol{x}_{\mathrm{r},i+1}^{\{k\}}\\ \vdots \\ \boldsymbol{x}_{\mathrm{r},i+n_\mathrm{h}+1}^{\{k\}}\end{bmatrix} = \underbrace{\begin{bmatrix}\boldsymbol{0}_{nn_\mathrm{h}\times n} & \boldsymbol{I}_{nn_\mathrm{h}}\\ \boldsymbol{0}_{n\times n} & \boldsymbol{0}_{n\times nn_\mathrm{h}}\end{bmatrix}}_{=:\boldsymbol{D}} \underbrace{\begin{bmatrix}\boldsymbol{x}_{\mathrm{r},i}^{\{k\}}\\ \vdots \\ \boldsymbol{x}_{\mathrm{r},i+n_\mathrm{h}}^{\{k\}}\end{bmatrix}}_{=:\boldsymbol{x}_{\mathrm{r},i:i+n_\mathrm{h}}^{\{k\}}}, \quad i = k, k+1, \dots, \tag{3.83}$$

wobei $\boldsymbol{0}_{n_1\times n_2}$ eine $n_1 \times n_2$ Nullmatrix und $\boldsymbol{I}_{nn_\mathrm{h}}$ eine $nn_\mathrm{h} \times nn_\mathrm{h}$ Einheitsmatrix bezeichnet, ausgedrückt werden. Der erweiterte Systemzustand

$$\tilde{\boldsymbol{x}}_{k,i} := \begin{bmatrix}\boldsymbol{x}_i^\mathsf{T} & \boldsymbol{x}_{\mathrm{r},i}^{\{k\}\mathsf{T}} & \dots & \boldsymbol{x}_{\mathrm{r},i+n_\mathrm{h}}^{\{k\}\mathsf{T}}\end{bmatrix}^\mathsf{T}, \quad i = k, k+1, \dots, \tag{3.84}$$

[61] Falls beschränkte Einschrittkosten r_i vorliegen und diese für $i \to \infty$ zudem verschwinden, ist das Kostenfunktional J_k selbst für $\gamma = 1$ endlich. Im Fall nicht-verschwindender aber beschränkter Einschrittkosten r_i sorgt $\gamma < 1$ dafür, dass J_k endlich ist (vgl. [SB18, Abschnitt 3.3]). Nicht-verschwindende Einschrittkosten r_i treten beispielsweise auf, wenn Stellenergie benötigt wird, um das System im Zustand $\boldsymbol{0}$ zu halten.

folgt somit der durch

$$\tilde{\boldsymbol{x}}_{k,i+1} = \begin{bmatrix} \boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{B}\boldsymbol{u}_i \\ \boldsymbol{D}\boldsymbol{x}^{\{k\}}_{\mathrm{r},i:i+n_\mathrm{h}} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{A} & \boldsymbol{0}_{n\times n(n_\mathrm{h}+1)} \\ \boldsymbol{0}_{n(n_\mathrm{h}+1)\times n} & \boldsymbol{D} \end{bmatrix}}_{=:\tilde{\boldsymbol{A}}} \underbrace{\begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{x}^{\{k\}}_{\mathrm{r},i:i+n_\mathrm{h}} \end{bmatrix}}_{=:\tilde{\boldsymbol{x}}_{k,i}} + \underbrace{\begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{0}_{n(n_\mathrm{h}+1)\times p} \end{bmatrix}}_{=:\tilde{\boldsymbol{B}}} \boldsymbol{u}_i \tag{3.85}$$

gegebenen Dynamik. Zudem kann $\boldsymbol{e}_i^\mathsf{T}\boldsymbol{Q}\boldsymbol{e}_i$ in (3.82) durch

$$\boldsymbol{e}_i^\mathsf{T}\boldsymbol{Q}\boldsymbol{e}_i = \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{x}^{\{k\}}_{\mathrm{r},i} \\ \vdots \\ \boldsymbol{x}^{\{k\}}_{\mathrm{r},i+n_\mathrm{h}} \end{bmatrix}^\mathsf{T} \underbrace{\begin{bmatrix} \boldsymbol{Q} & -\boldsymbol{Q} & \boldsymbol{0}_{n\times nn_\mathrm{h}} \\ -\boldsymbol{Q} & \boldsymbol{Q} & \boldsymbol{0}_{n\times nn_\mathrm{h}} \\ \boldsymbol{0}_{nn_\mathrm{h}\times n} & \boldsymbol{0}_{nn_\mathrm{h}\times n} & \boldsymbol{0}_{nn_\mathrm{h}\times nn_\mathrm{h}} \end{bmatrix}}_{=:\tilde{\boldsymbol{Q}}} \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{x}^{\{k\}}_{\mathrm{r},i} \\ \vdots \\ \boldsymbol{x}^{\{k\}}_{\mathrm{r},i+n_\mathrm{h}} \end{bmatrix} = \tilde{\boldsymbol{x}}_{k,i}^\mathsf{T}\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{x}}_{k,i} \tag{3.86}$$

ausgedrückt werden. Folglich lässt sich das durch (3.80)–(3.82) beschriebene Optimierungsproblem als diskontiertes Standard-LQ-Optimierungsproblem mit γ, $\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{B}}$, $\tilde{\boldsymbol{Q}}$ und $\boldsymbol{R}$ ausdrücken. Die gesuchte Aussage folgt analog zum Beweis von Satz 3.1. Hierbei wird genutzt, dass alle Eigenwerte von $\sqrt{\gamma}\boldsymbol{D}$ im Koordinatenursprung und somit innerhalb des Einheitskreises liegen. □

Bemerkung 3.6

Eine mögliche Alternative zur Annahme $\boldsymbol{x}^{\{k\}}_{\mathrm{r},i} = \boldsymbol{0}$, $\forall i > k + n_\mathrm{h}$, in (3.81) ist, anzunehmen, dass der Sollzustand jenseits des Vorausschauhorizonts der Länge n_h auf einen konstanten Wert $\neq \boldsymbol{0}$ gesetzt wird. Falls beispielsweise $\boldsymbol{x}^{\{k\}}_{\mathrm{r},i} = \boldsymbol{x}^{\{k\}}_{\mathrm{r},k+n_\mathrm{h}}$, $\forall i > k + n_\mathrm{h}$, gewählt wird, folgt für $\boldsymbol{D}$ in (3.83)

$$\boldsymbol{D} \coloneqq \begin{bmatrix} \boldsymbol{0}_{n(n_\mathrm{h}-1)\times n} & \boldsymbol{I}_{n(n_\mathrm{h}-1)} & \boldsymbol{0}_{n(n_\mathrm{h}-1)\times n} \\ \boldsymbol{0}_{n\times n} & \boldsymbol{0}_{n\times n(n_\mathrm{h}-1)} & \boldsymbol{I}_n \\ \boldsymbol{0}_{n\times n} & \boldsymbol{0}_{n\times n(n_\mathrm{h}-1)} & \boldsymbol{I}_n \end{bmatrix}. \tag{3.87}$$

Für $\gamma < 1$ bleibt die Aussage von Proposition 3.3 dann weiterhin bestehen. Die Lösungen dieser beiden Formulierungen nähern sich für kleiner werdende Diskontierungen γ und große Vorausschauhorizonte n_h einander an.

Bemerkung 3.7

Der Beweis von Proposition 3.3 offenbart, dass die Solltrajektoriendarstellung ADP-kompatibel gemäß Definition 3.1 ist. Hierbei ist

$$\boldsymbol{\zeta}_k \coloneqq \begin{bmatrix} \boldsymbol{x}_{\mathrm{r},k}^{\{k\}\mathsf{T}} & \dots & \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}}^{\{k\}\mathsf{T}} \end{bmatrix}^\mathsf{T}, \tag{3.88}$$

außerdem gilt

$$\boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k) \coloneqq \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0}_{n\times n_\mathrm{h}} \end{bmatrix} \boldsymbol{\zeta}_k \tag{3.89}$$

(vgl. (3.6)*) und analog zu* (3.7)

$$\boldsymbol{f}_{\boldsymbol{\zeta}}(\boldsymbol{\zeta}_k) \coloneqq \boldsymbol{D}\boldsymbol{\zeta}_k. \tag{3.90}$$

Der optimale Regler, der nach Proposition 3.3 existiert, soll, wie in der folgenden Problemdefinition zusammengefasst, ohne Kenntnis der Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ ermittelt werden.

Problem 3.3

Sei $\boldsymbol{u}_k^*, \boldsymbol{u}_{k+1}^*, \dots$ die Stellgrößensequenz, die das Gütefunktional J_k in (3.82) *für ein durch* (3.80) *beschriebenes System minimiert. Zudem seien die Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ unbekannt. Finde $\boldsymbol{u}_k^* = \boldsymbol{\mu}^*(\tilde{\boldsymbol{x}}_k)$, d. h. den optimalen Regler in Abhängigkeit von $\boldsymbol{x}_k$ und $\boldsymbol{x}_{\mathrm{r,soll},k}, \boldsymbol{x}_{\mathrm{r,soll},k+1}, \dots, \boldsymbol{x}_{\mathrm{r,soll},k+n_\mathrm{h}}$.*

3.3.2 Solltrajektorienabhängige Q-Function

Die zugrunde liegende Idee zur Lösung von Problem 3.3 ist die Definition einer solltrajektorienabhängigen Q-Function, d. h. einer Zustands-Aktions-Solltrajektorien-Nutzenfunktion im Gegensatz zur üblichen Zustands-Aktions-Nutzenfunktion. Diese Q-Function wird derart konstruiert, dass die sie minimierende Stellgröße eine Lösung für Problem 3.3 darstellt.

Im vorliegenden Abschnitt wird diese solltrajektorienabhängige Q-Function zunächst definiert, sowie deren analytische Lösung hergeleitet. Die analytische Lösung liefert wichtige Einblicke und beantwortet insbesondere die Frage einer geeigneten und möglichst effizienten Funktionsapproximation der Q-Function. Diese Funktionsapproximation wird in Abschnitt 3.3.3 für einen modellfreien ADP-Ansatz benötigt, der den optimalen Regler ohne Kenntnis der Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ lernt.

Zunächst werde der Fall eines endlichen Optimierungshorizonts der Länge K betrachtet, später erfolgt $K \to \infty$[62]. Die solltrajektorienabhängige Q-Function für einen Optimierungshorizont der Länge K sei wie nachfolgend gegeben definiert.

[62] Bei der nachfolgenden Notation ist eine Unterscheidung des Optimierungshorizonts K sowie des Vorausschauhorizonts n_h des Solltrajektorienverlaufs wichtig. Zudem muss zwischen dem aktuellen Zeitschritt k, in dem sich das System befindet und von dem ausgehend das Optimierungsproblem gelöst werden soll, und der Zeit κ auf dem Optimierungshorizont der Länge K unterschieden werden.

Definition 3.6 (Solltrajektorienabhängige Q-Function)
Die solltrajektorienabhängige Q-Function für einen Optimierungshorizont der Länge K mit der Kurzschreibweise ${}^{K}Q_{\kappa} \coloneqq Q_{K-\kappa}\left(\boldsymbol{x}_{k+\kappa}, \boldsymbol{u}_{k+\kappa}, \boldsymbol{x}_{\mathrm{r},k+\kappa}, \ldots, \boldsymbol{x}_{\mathrm{r},k+K}\right)$ sei durch die Bellman-Gleichung

$$\begin{aligned} {}^{K}Q_{\kappa} &\coloneqq r_{k+\kappa} + \gamma \sum_{i=k+\kappa+1}^{k+K} \gamma^{i-(k+\kappa+1)}\, r_i\big|_{\boldsymbol{u}_i^*} \\ &= r_{k+\kappa} + \gamma\; {}^{K}Q_{\kappa+1}\big|_{\boldsymbol{u}_{k+\kappa+1}^*} \end{aligned} \tag{3.91}$$

mit

$${}^{K}Q_{K} \coloneqq r\left(\boldsymbol{x}_{k+K}, \boldsymbol{u}_{k+K}, \boldsymbol{x}_{\mathrm{r},k+K}\right) \tag{3.92}$$

und $r_i \coloneqq r\left(\boldsymbol{x}_i, \boldsymbol{u}_i, \boldsymbol{x}_{\mathrm{r},i}\right)$ wie in (3.82) definiert. Dabei bezeichnet $\kappa \in \mathbb{N}_{\geq 0}$, $\kappa < K$, die Zeit auf dem aktuellen Optimierungshorizont der Länge K, beginnend bei k. Zudem gelte $\boldsymbol{x}_{\mathrm{r},i} \coloneqq \boldsymbol{x}_{\mathrm{r},i}^{\{k\}} = \boldsymbol{0}$, $\forall i > k + n_{\mathrm{h}}$ (siehe (3.81)).[63]

Folglich beschreibt ${}^{K}Q_{\kappa}$ die akkumulierten diskontierten Kosten vom Zeitschritt $k+\kappa$ bis zum Zeitschritt $k+K$, wenn die Stellgröße $\boldsymbol{u}_{k+\kappa}$ im Zeitschritt $k+\kappa$ und anschließend die optimalen Stellgrößen $\boldsymbol{u}_{k+\kappa+1}^*, \ldots, \boldsymbol{u}_{k+K}^*$, welche die Kosten auf dem verbliebenen Horizont minimieren, verwendet werden. Dabei ist der Sollzustandsverlauf wie in (3.81) auf einem gleitenden Vorausschauhorizont der Länge n_{h} gegeben. Für das durch

$${}^{K}J_k = \sum_{i=k}^{k+K} \gamma^{i-k} \underbrace{\frac{1}{2}\left(\boldsymbol{e}_i^{\intercal} \boldsymbol{Q} \boldsymbol{e}_i + \boldsymbol{u}_i^{\intercal} \boldsymbol{R} \boldsymbol{u}_i\right)}_{=\boldsymbol{r}_i} \tag{3.93}$$

definierte Kostenfunktional auf einem Optimierungshorizont der Länge K gilt das folgende Lemma.

Lemma 3.3
Die Stellgröße $\boldsymbol{u}_k$, welche die solltrajektorienabhängige Q-Function ${}^{K}Q_0$ minimiert, stellt eine optimale Lösung $\boldsymbol{u}_k^*$ dar, die ${}^{K}J_k$ minimiert.

[63] Die Notation $r_i|_{\boldsymbol{u}_i^*}$ meint, dass die optimale Stellgröße $\boldsymbol{u}_i^*$ in die Funktion $r(\cdot)$ nach (3.82) eingesetzt wird. Analog folgt ${}^{K}Q_{\kappa+1}\big|_{\boldsymbol{u}_{k+\kappa+1}^*}$ durch die Verwendung von $\boldsymbol{u}_{k+\kappa+1} = \boldsymbol{u}_{k+\kappa+1}^*$ in ${}^{K}Q_{\kappa+1}$.

Beweis:
Aus (3.93) und

$$\begin{aligned}\min_{\boldsymbol{u}_k} {}^K Q_0 &= r_k\big|_{\boldsymbol{u}_k^*} + \gamma\ {}^K Q_1\big|_{\boldsymbol{u}_{k+1}^*} \\ &= {}^K Q_0\big|_{\boldsymbol{u}_k^*} = \sum_{i=k}^{k+K} \gamma^{i-k}\ r_i\big|_{\boldsymbol{u}_i^*}\end{aligned} \tag{3.94}$$

folgt

$$\min_{\boldsymbol{u}_k,\dots,\boldsymbol{u}_{k+K}} {}^K J_k = \sum_{i=k}^{k+K} \gamma^{i-k}\ r_i\big|_{\boldsymbol{u}_i^*} = {}^K Q_0\big|_{\boldsymbol{u}_k^*}\,. \tag{3.95}$$

□

Für den Grenzübergang

$$\lim_{K\to\infty} {}^K J_k = J_k \tag{3.96}$$

folgt J_k nach (3.82). Mit der Q-Function ${}^K Q_0$ nach Definition 3.6 und als Resultat von Lemma 3.3 lässt sich Problem 3.3 daher in die folgende äquivalente Problemstellung umformulieren.

Problem 3.4
Seien die Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ unbekannt. Im Zeitschritt k sei das System im Zustand $\boldsymbol{x}_k$, zudem sei der Sollzustand auf einem Horizont der Länge n_h durch $\boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{x}_{\mathrm{r},k+1}, \dots, \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}}$ gegeben. Finde $\boldsymbol{u}_k^* = \boldsymbol{\mu}^*\left(\tilde{\boldsymbol{x}}_k\right)$, sodass die Q-Function

$$Q_0 \coloneqq \lim_{K\to\infty} {}^K Q_0 \tag{3.97}$$

minimiert wird.

Zunächst werde angenommen, dass die Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ bekannt sind, um die analytische Lösung von ${}^K Q_0$, wie im nachfolgenden Satz 3.4 gezeigt, zu untersuchen. Im nächsten Schritt wird dann eine iterative ADP-Methode verwendet, um ohne Kenntnis von $\boldsymbol{A}$ und $\boldsymbol{B}$, basierend auf dem TD-Fehler, die optimale Q-Function und das optimale Regelgesetz zu ermitteln.

Satz 3.4 (Analytische Lösung von ${}^{K}Q_0$)
Für $K \geq n_\mathrm{h}$ ist die zum Gütefunktional ${}^{K}J_k$ in (3.93) gehörende Q-Function ${}^{K}Q_0$ (vgl. Definition 3.6) durch

$$
{}^{K}Q_0 = \frac{1}{2}\begin{bmatrix}\boldsymbol{x}_k^\mathsf{T} & \boldsymbol{u}_k^\mathsf{T} & \boldsymbol{x}_{\mathrm{r},k}^\mathsf{T} & \cdots & \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}}^\mathsf{T} & \boldsymbol{0}^\mathsf{T}\end{bmatrix} \boldsymbol{H}_K \begin{bmatrix}\boldsymbol{x}_k \\ \boldsymbol{u}_k \\ \boldsymbol{x}_{\mathrm{r},k} \\ \vdots \\ \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}} \\ \boldsymbol{0}\end{bmatrix} \tag{3.98}
$$

gegeben. Hierbei sei $\boldsymbol{H}_K = \boldsymbol{H}_K^\mathsf{T} \in \mathbb{R}^{((K+2)n+p)\times((K+2)n+p)}$ mit

$$
\boldsymbol{H}_K = \begin{bmatrix}
\boldsymbol{h}_{\mathrm{xx}} & \boldsymbol{h}_{\mathrm{xu}} & \boldsymbol{h}_{\mathrm{xx}_{\mathrm{r},0}} & \boldsymbol{h}_{\mathrm{xx}_{\mathrm{r},1}} & \boldsymbol{h}_{\mathrm{xx}_{\mathrm{r},2}} & \cdots & \boldsymbol{h}_{\mathrm{xx}_{\mathrm{r},K}} \\
\boldsymbol{h}_{\mathrm{ux}} & \boldsymbol{h}_{\mathrm{uu}} & \boldsymbol{0} & \boldsymbol{h}_{\mathrm{ux}_{\mathrm{r},1}} & \boldsymbol{h}_{\mathrm{ux}_{\mathrm{r},2}} & \cdots & \boldsymbol{h}_{\mathrm{ux}_{\mathrm{r},K}} \\
\boldsymbol{h}_{\mathrm{x}_{\mathrm{r},0}\mathrm{x}} & \boldsymbol{0} & \boldsymbol{h}_{\mathrm{x}_{\mathrm{r},0}\mathrm{x}_{\mathrm{r},0}} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\
\boldsymbol{h}_{\mathrm{x}_{\mathrm{r},1}\mathrm{x}} & \boldsymbol{h}_{\mathrm{x}_{\mathrm{r},1}\mathrm{u}} & \boldsymbol{0} & \boldsymbol{h}_{\mathrm{x}_{\mathrm{r},1}\mathrm{x}_{\mathrm{r},1}} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\
\boldsymbol{h}_{\mathrm{x}_{\mathrm{r},2}\mathrm{x}} & \boldsymbol{h}_{\mathrm{x}_{\mathrm{r},2}\mathrm{u}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{h}_{\mathrm{x}_{\mathrm{r},2}\mathrm{x}_{\mathrm{r},2}} & \cdots & \boldsymbol{h}_{\mathrm{x}_{\mathrm{r},2}\mathrm{x}_{\mathrm{r},K}} \\
\boldsymbol{h}_{\mathrm{x}_{\mathrm{r},3}\mathrm{x}} & \boldsymbol{h}_{\mathrm{x}_{\mathrm{r},3}\mathrm{u}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{h}_{\mathrm{x}_{\mathrm{r},3}\mathrm{x}_{\mathrm{r},2}} & \cdots & \boldsymbol{h}_{\mathrm{x}_{\mathrm{r},3}\mathrm{x}_{\mathrm{r},K}} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\boldsymbol{h}_{\mathrm{x}_{\mathrm{r},K}\mathrm{x}} & \boldsymbol{h}_{\mathrm{x}_{\mathrm{r},K}\mathrm{u}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{h}_{\mathrm{x}_{\mathrm{r},K}\mathrm{x}_{\mathrm{r},2}} & \cdots & \boldsymbol{h}_{\mathrm{x}_{\mathrm{r},K}\mathrm{x}_{\mathrm{r},K}}
\end{bmatrix}. \tag{3.99}
$$

Die genauen Werte von $\boldsymbol{H}_K$ folgen aus den Berechnungen im nachfolgenden Beweis.

Beweis:
Der Beweis ist in Anhang A.1 skizziert. □

Für $K \to \infty$ sei $\boldsymbol{H}$ die nordwestliche $((n_\mathrm{h}+2)n+p)\times((n_\mathrm{h}+2)n+p)$-Teilmatrix von $\boldsymbol{H}_K$. Dann folgt wegen $\boldsymbol{x}_{\mathrm{r},i} = \boldsymbol{0}$, $\forall i > k + n_\mathrm{h}$,

$$
Q_0 = \lim_{K\to\infty} {}^{K}Q_0 = \frac{1}{2}\tilde{\boldsymbol{y}}_k^\mathsf{T}\boldsymbol{H}\tilde{\boldsymbol{y}}_k, \tag{3.100}
$$

wobei $\tilde{\boldsymbol{y}}_k := \begin{bmatrix}\boldsymbol{x}_k^\mathsf{T} & \boldsymbol{u}_k^\mathsf{T} & \boldsymbol{x}_{\mathrm{r},k}^\mathsf{T} & \cdots & \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}}^\mathsf{T}\end{bmatrix}^\mathsf{T}$.

Daher ist die Q-Function des Solltrajektorienfolgeregelungsproblems (vgl. Problem 3.3 bzw. Problem 3.4) quadratisch bezüglich des Systemzustands $\boldsymbol{x}_k$, der Stellgröße $\boldsymbol{u}_k$ und des Sollzustandsverlaufs $\boldsymbol{x}_{\mathrm{r},k}, \ldots, \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}}$ und zudem vollständig durch die Matrix $\boldsymbol{H}$ parametriert. Somit stellt die hier präsentierte Q-Function Q_0 eine Verallgemeinerung der aus dem Regulationsfall [BYB94], [Lan97] bekannten Struktur der Q-Function dar. Weiterhin ist Q_0 nicht nur quadratisch, vielmehr weist $\boldsymbol{H}$ eine spezielle Struktur auf. Die Relevanz der aus Satz 3.4 folgenden Struktur ist in der nachfolgenden Bemerkung zusammengefasst.

Bemerkung 3.8

Satz 3.4 liefert Aussagen über die genaue analytische Lösung der vorgestellten sollzustandsabhängigen Q-Function. Die damit einhergehende Erkenntnis über die exakte Struktur der Q-Function ermöglicht eine effiziente Wahl eines geeigneten Funktionsapproximators und erlaubt somit eine signifikante Reduktion der benötigten zu lernenden Gewichte (vgl. Lemma 3.4 und Lemma 3.5 sowie Abschnitt 3.3.5) bei der späteren ADP-Umsetzung.

3.3.3 Modellfreies Erlernen der Q-Function

Im Folgenden wird die Q-Function, die explizit vom Sollzustandsverlauf auf einem gleitenden Vorausschauhorizont der Länge n_h abhängt, ohne Kenntnis der Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ gelernt. Konkret wird dabei $\boldsymbol{H}$ (vgl. (3.100)) aus Zustandsübergängen und damit verknüpften Einschrittkosten ermittelt.

Die optimale Stellgröße $\boldsymbol{u}_k^* = \boldsymbol{\mu}^*\left(\tilde{\boldsymbol{x}}_k\right)$, die für $\kappa = 0$ und $K \to \infty$ äquivalent zu (A.17) ist, kann dann, wie im nachfolgenden Korollar gegeben, direkt aus $\boldsymbol{H}$ bestimmt werden. Dabei stellt (A.18) sicher, dass $\boldsymbol{u}_k^*$ tatsächlich die Q-Function Q_0 minimiert.

Korollar 3.1 (Optimaler Regler in Abhängigkeit von $\boldsymbol{H}$)

Aus Lemma 3.3 und Satz 3.4 folgt direkt, dass die nach Problem 3.4 optimale Stellgröße im Zeitschritt k durch

$$\begin{aligned}\boldsymbol{u}_k^* = \boldsymbol{\mu}^*\left(\tilde{\boldsymbol{x}}_k\right) &= -\boldsymbol{h}_\mathrm{uu}^{-1}\begin{bmatrix}\boldsymbol{h}_\mathrm{ux} & \boldsymbol{h}_{\mathrm{ux}_{\mathrm{r},1}} & \cdots & \boldsymbol{h}_{\mathrm{ux}_{\mathrm{r},n_\mathrm{h}}}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_k\\ \boldsymbol{x}_{\mathrm{r},k+1}\\ \vdots\\ \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}}\end{bmatrix}\\ &=: -\underbrace{\begin{bmatrix}\boldsymbol{K}_\mathrm{x}^* & \boldsymbol{K}_\mathrm{ref}^*\end{bmatrix}}_{=:\boldsymbol{K}^*}\begin{bmatrix}\boldsymbol{x}_k\\ \boldsymbol{x}_{\mathrm{r},k+1}\\ \vdots\\ \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}}\end{bmatrix}\end{aligned} \tag{3.101}$$

gegeben ist.

Im Gegensatz zu bestehenden ADP-Reglern (vgl. Abschnitt 2.2) bezieht das optimale Regelgesetz (3.101) den exakten Sollzustandsverlauf $\boldsymbol{x}_{\mathrm{r},k+1}, \ldots, \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}}$ auf einem Vorausschauhorizont der Länge n_h explizit ein und ist daher für flexible Solltrajektorienverläufe geeignet.

Bemerkung 3.9

Nach [Kuč72, Theorem 8] folgt, dass das optimale Regelgesetz $\boldsymbol{\mu}^*(\tilde{\boldsymbol{x}}_k)$ aus Proposition 3.3 das fiktive System

$$\tilde{\boldsymbol{x}}_{k+1} = \sqrt{\gamma}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}_k + \sqrt{\gamma}\tilde{\boldsymbol{B}}\boldsymbol{\mu}^*(\tilde{\boldsymbol{x}}_k) \tag{3.102}$$

stabilisiert. Im Fall $\gamma = 1$ folgt daraus direkt, dass das geregelte System

$$\boldsymbol{x}_{k+1} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_{\mathrm{x}}^*)\,\boldsymbol{x}_k - \boldsymbol{B}\boldsymbol{K}_{\mathrm{ref}}^* \begin{bmatrix} \boldsymbol{x}_{\mathrm{r},k+1} \\ \vdots \\ \boldsymbol{x}_{\mathrm{r},k+n_{\mathrm{h}}} \end{bmatrix} \tag{3.103}$$

für beschränkte Sollzustandsvorgaben $\boldsymbol{x}_{\mathrm{r},k+1}, \dots, \boldsymbol{x}_{\mathrm{r},k+n_{\mathrm{h}}}$ Referenz-Zustands-stabil nach Definition 3.5 ist. Für $\gamma < 1$ muss, analog zu Abschnitt 3.2.4, überprüft werden, ob sämtliche Eigenwerte von $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_{\mathrm{x}}^*$ im Inneren des Einheitskreises liegen. Falls $\boldsymbol{A}$ und $\boldsymbol{B}$ nicht explizit bekannt sind, kann hierzu anhand der Matrix $\boldsymbol{H}$ aus (3.99) *das durch Satz 3.3 gegebene hinreichende Kriterium verwendet werden.*

3.3.3.1 Funktionsapproximation der erweiterten Q-Function

Um die Q-Function datenbasiert und ohne explizite Kenntnis der Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ zu lernen, wird Q_0 durch einen Funktionsapproximator (vgl. Abschnitt 2.1.3) parametriert. Anschließend wird der quadratische TD-Fehler genutzt, um eine Value Iteration (vgl. Abschnitt 2.1.4.2) durchzuführen, deren Resultat die geschätzten optimalen Q-Function-Gewichte sowie das zugehörige optimale Regelgesetz darstellt. Hierzu sei die geschätzte Q-Function $\hat{Q}_0$ durch eine Summe gewichteter Basisfunktionen gegeben:

$$\hat{Q}_0 = \hat{\boldsymbol{w}}^\intercal \boldsymbol{\phi}\left(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{x}_{\mathrm{r},k}, \dots, \boldsymbol{x}_{\mathrm{r},k+n_{\mathrm{h}}}\right) = \hat{\boldsymbol{w}}^\intercal \boldsymbol{\phi}(\tilde{\boldsymbol{y}}_k). \tag{3.104}$$

Dabei stellt $\hat{\boldsymbol{w}} \in \mathbb{R}^h$ den zu schätzenden Gewichtsvektor dar und $\boldsymbol{\phi} : \mathbb{R}^{(n_{\mathrm{h}}+2)n+p} \to \mathbb{R}^h$ ist ein Basisfunktionsvektor. Im Gegensatz zu klassischen Q-Functions bezieht $\hat{Q}_0$ in (3.104) somit explizit den durch $\boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{x}_{\mathrm{r},k+1}, \dots, \boldsymbol{x}_{\mathrm{r},k+n_{\mathrm{h}}}$ beschriebenen Solltrajektorienverlauf ein. Das nachfolgende Lemma besagt, dass die Q-Function Q_0 mithilfe quadratischer Basisfunktionen exakt durch $\hat{Q}_0$ repräsentiert werden kann.

Lemma 3.4

Mit

$$h = \frac{1}{2}\big((n_{\mathrm{h}} + 2)n + p\big)\big((n_{\mathrm{h}} + 2)n + p + 1\big) - \big(n^2(2n_{\mathrm{h}} - 1) + pn\big) \tag{3.105}$$

quadratischen Basisfunktionen $\boldsymbol{\phi} = \begin{bmatrix}\phi_1 & \cdots & \phi_h\end{bmatrix}^\intercal$ existiert ein optimaler Gewichtsvektor $\hat{\boldsymbol{w}} = \boldsymbol{w}^*$, sodass $\hat{Q}_0 = Q_0$ gilt.

Beweis:
Aufgrund der Symmetrie von $\boldsymbol{H}_K$ in (3.99) und der darin auftauchenden Nullen weist $\boldsymbol{H}$ maximal h nicht-redundante Elemente auf, wobei h durch (3.105) gegeben ist. Für $\bar{h} = 1, \ldots, h$ werden quadratische Basisfunktionen, deren Elemente die Form

$$\phi_{\bar{h}} = \begin{cases} \tilde{y}_{k,i}\tilde{y}_{k,j}, & \text{für } i \neq j \\ \frac{1}{2}\tilde{y}_{k,i}^2, & \text{für } i = j, \end{cases} \tag{3.106}$$

aufweisen, definiert. Hierbei indiziere i, j die entsprechenden nicht-redundanten Elemente von $\boldsymbol{H}$, $\tilde{\boldsymbol{y}}_k$ sei wie in (3.100) definiert und $\tilde{y}_{k,i}$ bezeichne das i-te Element des Vektors $\tilde{\boldsymbol{y}}_k$. Folglich ist Q_0 in (3.100) gerade dann äquivalent zu $\hat{Q}_0$ in (3.104), wenn die Gewichte $\hat{w}_i$, $i = 1, \ldots, h$, identisch zu den entsprechenden nicht-redundanten Elementen von $\boldsymbol{H}$ sind, d. h. wenn $\hat{\boldsymbol{w}} = \boldsymbol{w}^*$ gilt. □

Somit gibt h in Lemma 3.4 die maximal benötigte Anzahl an Basisfunktionen und zugehörigen zu lernenden Gewichten an, um Q_0 exakt parametrieren zu können. In der Anwendung liegen jedoch häufig dünnbesetzte Gewichtsmatrizen $\boldsymbol{Q}$ im Gütefunktional (3.82) vor. Daher kann die Anzahl der benötigten Basisfunktionen und somit die Dimension von $\hat{\boldsymbol{w}}$ weiter reduziert werden. Hierfür wird die aus Satz 3.4 und dem zugehörigen Beweis resultierende Struktur von $\boldsymbol{H}$, die auch ohne explizite Kenntnis der Zahlenwerte in $\boldsymbol{A}$ und $\boldsymbol{B}$ exakt bekannt ist, genutzt. Dies wird im folgenden Lemma näher ausgeführt.

Lemma 3.5
Falls die o-te Zeile und o-te Spalte von $\boldsymbol{Q}$ null ist, dann sind die

- *o-te Spalte von $\boldsymbol{h}_{\mathrm{xx}_{\mathrm{r},i}} \forall i \in \{0, \ldots, n_\mathrm{h}\}$,*
- *o-te Spalte von $\boldsymbol{h}_{\mathrm{ux}_{\mathrm{r},i}} \forall i \in \{0, \ldots, n_\mathrm{h}\}$ und*
- *o-te Zeile und o-te Spalte von $\boldsymbol{h}_{\mathrm{x}_{\mathrm{r},i}\mathrm{x}_{\mathrm{r},j}} \forall i, j \in \{0, \ldots, n_\mathrm{h}\}$*

ebenfalls null. Folglich reduziert sich die Anzahl h der nicht-redundanten Gewichte in $\boldsymbol{H}$, und somit die Dimension des zu schätzenden Gewichtsvektors $\hat{\boldsymbol{w}}$, zu

$$\begin{aligned} h = {} & (n-q)(n-q+1)\left(\frac{n_\mathrm{h}}{2}+1\right) + \frac{1}{2}n(n+1) \\ & + \big(p + n_\mathrm{h}(n-q)\big)(p+n) + \frac{1}{2}(n_\mathrm{h}-2)(n_\mathrm{h}-1)(n-q)^2. \end{aligned} \tag{3.107}$$

Hierbei wurde $\boldsymbol{h}_{\mathrm{xx}_{\mathrm{r},0}} = \boldsymbol{Q}$ und $\boldsymbol{h}_{\mathrm{x}_{\mathrm{r},0}\mathrm{x}_{\mathrm{r},0}} = -\boldsymbol{Q}$ berücksichtigt, weiterhin bezeichnet q die Anzahl der Zeilen bzw. Spalten von $\boldsymbol{Q}$, die null sind.

Beweis:
Die Aussage folgt direkt aus Betrachtung von (A.15) unter Berücksichtigung von (A.11)–(A.14). □

Obwohl Lemma 3.5 sehr technischer Natur ist, ist die Kenntnis über die Dünnbesetztheit der Matrix $\boldsymbol{H}$ essenziell für die Implementierung effizienter ADP-Regler mit einer überschaubaren Anzahl h zu schätzender Elemente von $\hat{\boldsymbol{w}}$. Dank der genauen Kenntnis der spezifischen Struktur der analytischen Lösung von Q_0, wie sie in Satz 3.4 und dessen Beweis hergeleitet wurde, genügen somit nach Lemma 3.5 h quadratische Basisfunktionen mit h nach (3.107), um Q_0 exakt zu parametrieren (vgl. Beispiel 3.1).

Beispiel 3.1 (Reduzierung der Dimension h des Critic-Gewichts $\hat{\boldsymbol{w}}$)

Tabelle 3.2 illustriert die Reduzierung der Dimension h des Critic-Gewichts $\hat{\boldsymbol{w}}$ durch Lemma 3.4 bzw. Lemma 3.5 im Vergleich dazu, nur die Symmetrie der Matrix $\boldsymbol{H}$ zu berücksichtigen, für ein System zweiter Ordnung und ein System sechster Ordnung.

		System 1 ($n = 2, p = 1$)		*System 2* ($n = 6, p = 1$)	
	q	1	1	5	5
	n_{h}	10	100	10	100
$\boldsymbol{H}$ *symmetrisch*	h	325	21115	2701	188191
Lemma 3.4	h	247	20317	2011	181021
Lemma 3.5	h	84	5259	146	5681

Tabelle 3.2: Die Berücksichtigung der Struktur der Matrix $\boldsymbol{H}$ erlaubt bei den betrachteten Beispielsystemen eine signifikante Verringerung der Dimension h des zu schätzenden Gewichtsvektors $\hat{\boldsymbol{w}}$.

Bemerkung 3.10

Die zu lernenden Gewichte $\hat{\boldsymbol{w}}$ hängen nicht von der Solltrajektorie ab. Somit verdeutlichen (3.104) *und Lemma 3.5 auch, dass die vorgestellte Q-Function, die explizit vom Solltrajektorienverlauf abhängt, über beliebige Solltrajektorien, die auf einem Vorausschauhorizont der Länge n_{h} gegeben sind, generalisiert.*

3.3.3.2 Datenbasierter Lernalgorithmus der erweiterten Q-Function

In diesem Abschnitt wird der auf einer Value Iteration basierende ADP-Algorithmus vorgestellt, der die Gewichte $\hat{\boldsymbol{w}}$ basierend auf der Minimierung des quadrierten TD-Fehlers und ohne Kenntnis der Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ adaptiert. Hierzu sei zunächst

$$\begin{aligned}\hat{Q}_1 &:= \hat{\boldsymbol{w}}^{\mathsf{T}} \boldsymbol{\phi}\left(\boldsymbol{x}_{k+1}, \boldsymbol{u}_{k+1}, \boldsymbol{x}_{\mathrm{r},k+1}, \ldots, \boldsymbol{x}_{\mathrm{r},k+n_{\mathrm{h}}}, \boldsymbol{0}\right) \\ &= \hat{\boldsymbol{w}}^{\mathsf{T}} \boldsymbol{\phi}\left(\tilde{\boldsymbol{y}}_{k+1}\right)\end{aligned} \tag{3.108}$$

mit $\tilde{\boldsymbol{y}}_{k+1} = \begin{bmatrix}\boldsymbol{x}_{k+1}^\mathsf{T} & \boldsymbol{u}_{k+1}^\mathsf{T} & \boldsymbol{x}_{\mathrm{r},k+1}^\mathsf{T} & \cdots & \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}}^\mathsf{T} & \boldsymbol{0}^\mathsf{T}\end{bmatrix}^\mathsf{T}$ definiert[64]. Die geschätzte optimale Stellgröße $\hat{\boldsymbol{u}}_{k+\kappa}^*$, die auf der geschätzten Q-Function $\hat{Q}_\kappa\Big|_{\boldsymbol{u}_{k+\kappa}}$ basiert, werde zudem durch

$$\begin{aligned}\hat{\boldsymbol{u}}_{k+\kappa}^* &:= \arg\min_{\boldsymbol{u}_{k+\kappa}} \hat{Q}_\kappa\Big|_{\boldsymbol{u}_{k+\kappa}} \\ &= -\boldsymbol{K}\begin{bmatrix}\boldsymbol{x}_{k+\kappa}^\mathsf{T} & \boldsymbol{x}_{\mathrm{r},k+\kappa+1}^\mathsf{T} & \cdots & \boldsymbol{x}_{\mathrm{r},k+\kappa+n_\mathrm{h}}^\mathsf{T}\end{bmatrix}^\mathsf{T}\end{aligned} \tag{3.109}$$

mit $\boldsymbol{K} := \hat{\boldsymbol{h}}_\mathrm{uu}^{-1}\begin{bmatrix}\hat{\boldsymbol{h}}_\mathrm{ux} & \hat{\boldsymbol{h}}_{\mathrm{ux}_{\mathrm{r},1}} & \cdots & \hat{\boldsymbol{h}}_{\mathrm{ux}_{\mathrm{r},n_\mathrm{h}}}\end{bmatrix}$ (vgl. (3.101)) definiert. Dabei sind durch $\hat{\boldsymbol{h}}_{(\cdot)}$ analog zu (3.99) die Teilmatrizen von $\hat{\boldsymbol{H}}$ gegeben, wobei $\hat{\boldsymbol{H}}$ die auf $\hat{\boldsymbol{w}}$ basierende Schätzung der Matrix $\boldsymbol{H}$ bezeichnet. Somit resultiert $\hat{\boldsymbol{u}}_{k+\kappa}^* = \boldsymbol{u}_{k+\kappa}^*$, falls $\hat{\boldsymbol{w}} = \boldsymbol{w}^*$ gilt und die optimale Reglermatrix $\boldsymbol{K} = \boldsymbol{K}^*$ ergibt sich, wenn $\hat{\boldsymbol{H}} = \boldsymbol{H}$ gilt. Basierend auf der Bellman-Gleichung (3.91) wird nachfolgend der TD-Fehler δ_k (vgl. [Sut88]) definiert, der für $\hat{\boldsymbol{w}} = \boldsymbol{w}^*$ verschwindet.

Definition 3.7 (TD-Fehler der solltrajektorienabhängigen Q-Function)
Der TD-Fehler δ_k, d. h. der Approximationsfehler der Bellman-Gleichung (3.91)*, der aufgrund einer Abweichung des geschätzten Gewichts $\hat{\boldsymbol{w}}$ von $\boldsymbol{w}^*$ resultiert, sei durch*

$$\begin{aligned}\delta_k &:= r_k + \gamma\,\hat{Q}_1\Big|_{\hat{\boldsymbol{u}}_{k+1}^*} - \hat{Q}_0 \\ &= r_k + \gamma\hat{\boldsymbol{w}}^\mathsf{T}\phi\left(\tilde{\boldsymbol{y}}_{k+1}^*\right) - \hat{\boldsymbol{w}}^\mathsf{T}\phi\left(\tilde{\boldsymbol{y}}_k\right)\end{aligned} \tag{3.110}$$

definiert. Dabei sei $\tilde{\boldsymbol{y}}_{k+1}^* := \tilde{\boldsymbol{y}}_{k+1}\big|_{\boldsymbol{u}_{k+1}=\hat{\boldsymbol{u}}_{k+1}^*}$.

Um die Schätzung der sollzustandsabhängigen Q-Function $\hat{Q}_0$ sowie die daraus resultierende geschätzte optimale Stellgröße $\hat{\boldsymbol{u}}_k^*$ zu verbessern, wird im Folgenden eine Value Iteration (vgl. Abschnitt 2.1.4.2) verwendet. Im Policy-Evaluation-Schritt wird die Gewichtsschätzung $\hat{\boldsymbol{w}}^{[l]}$, die in der Iteration l die geschätzte Q-Function repräsentiert, angepasst. Im nachfolgenden Policy-Improvement-Schritt wird, ausgehend von der neuen Schätzung $\hat{\boldsymbol{w}}^{[l]}$ des Q-Function-Gewichts und der zugehörigen Matrix $\boldsymbol{H}^{[l]}$, das Regelgesetz $\boldsymbol{K}^{[l]}$, wie in (3.109) gezeigt, adaptiert. Zur Berechnung von δ_k in (3.110) wird $\hat{\boldsymbol{u}}_{k+1}^*$ benötigt, jedoch ist das optimale Gewicht $\boldsymbol{w}^*$ a priori unbekannt. Daher wird zunächst $\hat{\boldsymbol{w}}^{[0]} = \boldsymbol{0}$ und $\hat{\boldsymbol{u}}_{k+1}^{*[0]} = \boldsymbol{0}$ initialisiert. Letzteres geschieht durch die Wahl von $\boldsymbol{K}^{[0]} = \boldsymbol{0}$[65]. Im Policy-Evaluation-Schritt ist das Ziel,

[64] Für die in Bemerkung 3.6 beschriebene alternative Problemformulierung, welche $\boldsymbol{x}_{\mathrm{r},i}^{\{k\}} = \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}}^{\{k\}}$, $\forall i > k + n_\mathrm{h}$, annimmt, müsste an dieser Stelle $\boldsymbol{\phi}\left(\boldsymbol{x}_{k+1}, \boldsymbol{u}_{k+1}, \boldsymbol{x}_{\mathrm{r},k+1}, \ldots, \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}}, \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}}\right)$ verwendet und $\tilde{\boldsymbol{y}}_{k+1} = \begin{bmatrix}\boldsymbol{x}_{k+1}^\mathsf{T} & \boldsymbol{u}_{k+1}^\mathsf{T} & \boldsymbol{x}_{\mathrm{r},k+1}^\mathsf{T} & \cdots & \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}}^\mathsf{T} & \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}}^\mathsf{T}\end{bmatrix}^\mathsf{T}$ gesetzt werden.

[65] Bei vorhandenem Vorwissen kann die Initialisierung zugunsten einer schnelleren Konvergenz entsprechend angepasst werden.

eine neue Schätzung $\hat{\boldsymbol{w}}^{[l+1]}$ zu erlangen, sodass $\left(\delta_k^{[l]}\right)^2$ minimiert wird, wobei in Analogie zu (3.110)

$$\delta_k^{[l]} := r_k + \gamma \hat{\boldsymbol{w}}^{[l]\mathsf{T}} \boldsymbol{\phi}\left(\tilde{\boldsymbol{y}}_{k+1}^{*[l]}\right) - \hat{\boldsymbol{w}}^{[l+1]\mathsf{T}} \boldsymbol{\phi}\left(\tilde{\boldsymbol{y}}_k\right) \tag{3.111}$$

gesetzt wird. In (3.111) sei $\tilde{\boldsymbol{y}}_{k+1}^{*[l]} := \tilde{\boldsymbol{y}}_{k+1}\big|_{\boldsymbol{u}_{k+1}=\hat{\boldsymbol{u}}_{k+1}^{*[l]}}$, wobei $\hat{\boldsymbol{u}}_{k+1}^{*[l]}$ aus (3.109) mit $\boldsymbol{K} = \boldsymbol{K}^{[l]}$ folgt. Mit $\hat{\boldsymbol{w}}^{[l]} \in \mathbb{R}^h$ wie in Lemma 3.5 ergibt sich, dass $\delta_k^{[l]}$ mindestens zu $M \geq h$ unterschiedlichen Zeitschritten betrachtet werden muss, um eine Least-Squares-Schätzung unter Nutzung von M Datentupeln durchführen zu können. Damit resultiert

$$\hat{\boldsymbol{w}}^{[l+1]} = \underset{\hat{\boldsymbol{w}}^{[l+1]}}{\arg\min} \sum_{j=k-M+1}^{k} \left(\delta_j^{[l]}\right)^2. \tag{3.112}$$

Es seien weiterhin

$$\boldsymbol{\Phi} := \begin{bmatrix} \boldsymbol{\phi}\left(\tilde{\boldsymbol{y}}_{k-M+1}\right) & \dots & \boldsymbol{\phi}\left(\tilde{\boldsymbol{y}}_k\right) \end{bmatrix}^\mathsf{T} \tag{3.113}$$

und

$$\boldsymbol{r} := \begin{bmatrix} r_{k-M+1} + \gamma \hat{\boldsymbol{w}}^{[l]\mathsf{T}} \phi\left(\tilde{\boldsymbol{y}}_{k-M+2}^{*[l]}\right) \\ \vdots \\ r_k + \gamma \hat{\boldsymbol{w}}^{[l]\mathsf{T}} \phi\left(\tilde{\boldsymbol{y}}_{k+1}^{*[l]}\right) \end{bmatrix} \tag{3.114}$$

definiert. Falls die Anregungsbedingung

$$\operatorname{Rang}\left(\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi}\right) = h \tag{3.115}$$

erfüllt ist, existiert eine eindeutige Lösung $\hat{\boldsymbol{w}}^{[l+1]}$, die (3.112) minimiert. Diese ist durch

$$\hat{\boldsymbol{w}}^{[l+1]} = \left(\boldsymbol{\Phi}^\mathsf{T}\boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^\mathsf{T}\boldsymbol{r} \tag{3.116}$$

gegeben (vgl. [ÅW95, Theorem 2.1]).

Der Policy-Improvement-Schritt ergibt sich dann aus der neuen Gewichtsschätzung $\hat{\boldsymbol{w}}^{[l+1]}$ und der zugehörigen Matrix $\boldsymbol{H}^{[l+1]}$ zu

$$\boldsymbol{K}^{[l+1]} = \boldsymbol{h}_{\mathrm{uu}}^{[l+1]^{-1}} \begin{bmatrix} \boldsymbol{h}_{\mathrm{ux}}^{[l+1]} & \boldsymbol{h}_{\mathrm{ux}_{\mathrm{r},1}}^{[l+1]} & \dots & \boldsymbol{h}_{\mathrm{ux}_{\mathrm{r},n_\mathrm{h}}}^{[l+1]} \end{bmatrix} \tag{3.117}$$

(vgl. Korollar 3.1). Diese Iteration wird, bei festem Zeitschritt k, so oft wiederholt, bis sich $\hat{\boldsymbol{w}}^{[l]}$ nicht mehr signifikant ändert, d. h. $\left\|\hat{\boldsymbol{w}}^{[l]} - \hat{\boldsymbol{w}}^{[l-1]}\right\|_2$ einen Schwellwert $e_{\hat{\boldsymbol{w}}}$ unterschreitet.

Wenngleich die Q-Function zu der durch $\boldsymbol{K}^{[l]}$ gegebenen Target Policy $\hat{\boldsymbol{\mu}}^{*[l]}(\cdot)$, die in Form von $\hat{\boldsymbol{u}}_{k+1}^{*[l]}$ in $\tilde{\boldsymbol{y}}_{k+1}^{*[l]}$ nach (3.111) berücksichtigt wird, ausgewertet werden soll, wurde bislang noch nicht diskutiert, welche Behavior Policy verwendet wird (vgl. Abschnitt 2.1.4.4). Die Behavior Policy stellt dabei die Stellgröße $\boldsymbol{u}_k$ dar, die auf das System angewandt wird und welche

in den Ausdrücken r_k und $\tilde{\boldsymbol{y}}_k$ in (3.111) auftaucht. Während die Target Policy die aktuelle Schätzung des optimalen Regelgesetzes darstellt, kann die Behavior Policy genutzt werden, um die in (3.115) gegebene Anregungsbedingung zu erfüllen. Während der Adaptionsprozess aktiv ist, sei

$$\boldsymbol{u}_k = \tilde{\boldsymbol{u}}_k^{*[l]} := \hat{\boldsymbol{u}}_k^{*[l]} + \boldsymbol{\xi}_k. \tag{3.118}$$

Dabei dient $\boldsymbol{\xi}_k \sim \mathcal{N}\left(\boldsymbol{0}; \boldsymbol{\mathcal{V}}\right)$ als Anregungsrauschen, welches den Systemzustand $\boldsymbol{x}_k$ anregt. Zur Anregung des Sollzustandsverlaufs kann dieser ebenfalls durch additives Gaußsches Rauschen ξ_{ref} überlagert werden, um bei einem während des Trainings möglicherweise abwechslungsarmen Sollzustandsverlauf dennoch ausreichende Anregung sicherzustellen.

Der gesamte ADP-Algorithmus, der ohne Kenntnis der Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ lernt, einer auf einem gleitenden Vorausschauhorizont der Länge n_h gegebenen, beliebigen Solltrajektorie optimal im Sinne des Gütefunktionals (3.82) zu folgen, ist in Algorithmus 3.2 gegeben.

Algorithmus 3.2 ADP-Algorithmus mit Sollverlauf auf endlichem Vorausschauhorizont

Initialisiere M, $\hat{\boldsymbol{w}} := \hat{\boldsymbol{w}}^{[0]} := \boldsymbol{0}$, $\boldsymbol{K}^{[0]} := \boldsymbol{0}$
for $k = 0, 1, 2, \ldots$ **do**
  wende die Stellgröße $\tilde{\boldsymbol{u}}_k^{*[l]}$ nach (3.118) auf das System (3.80) an
  if $k + 1 \mod M = 0$ **then**
    $l := 0, \hat{\boldsymbol{w}}^{[l]} := \hat{\boldsymbol{w}}$
    do
      Policy-Evaluation-Schritt: berechne $\hat{\boldsymbol{w}}^{[l+1]}$ mit (3.112)
      Policy-Improvement-Schritt: berechne $\boldsymbol{K}^{[l+1]}$ mit (3.117)
      $l := l + 1$
    while $\left\| \hat{\boldsymbol{w}}^{[l]} - \hat{\boldsymbol{w}}^{[l-1]} \right\|_2 > e_{\hat{\boldsymbol{w}}}$
    $\hat{\boldsymbol{w}} := \hat{\boldsymbol{w}}^{[l]}$
  end if
end for

Bemerkung 3.11

Die in Algorithmus 3.2 verwendete iterative Methode aus Policy-Evaluation- und Policy-Improvement-Schritt stellt eine Value Iteration dar. Dies manifestiert sich in der Definition von δ_k in (3.111), wobei $\hat{Q}_1$ auf $\hat{\boldsymbol{w}}^{[l]}$ basiert und $\hat{Q}_0$ auf $\hat{\boldsymbol{w}}^{[l+1]}$ (vgl. (2.25)).
Weiterhin gehört der Algorithmus zur Klasse der Off-Policy-Methoden (vgl. Abschnitt 2.1.4.4), da die Behavior Policy $\tilde{\boldsymbol{u}}_k^{*[l]} = \hat{\boldsymbol{u}}_k^{*[l]} + \boldsymbol{\xi}_k$ angewandt wird, während die zur Target Policy $\hat{\boldsymbol{u}}_k^{*[l]}$ gehörende Q-Function gelernt wird.

3.3.3.3 Konvergenzanalyse des Lernalgorithmus

In diesem Abschnitt wird die Konvergenz der geschätzten solltrajektorienabhängigen Q-Function $\hat{Q}_0$ gegen die optimale Q-Function Q_0 untersucht. Konkret wird gezeigt, dass für $l \to \infty$ die Konvergenz des Gewichts $\hat{\boldsymbol{w}}^{[l]} \to \boldsymbol{w}^*$ und damit auch der Matrix $\boldsymbol{H}^{[l]} \to \boldsymbol{H}$ folgt. Somit konvergiert die Value Iteration für die erweiterte sollzustandsabhängige Q-Function gegen das optimale Regelgesetz, d. h. $\boldsymbol{K}^{[l]} \to \boldsymbol{K}^*$ folgt.

Die Konvergenzanalyse ist dabei wie folgt strukturiert: Zuerst wird gezeigt, dass die Value Iteration, d. h. die Iteration zwischen (3.112) und (3.117), äquivalent zu einer durch $\boldsymbol{H}^{[l]}$ gegebenen Matrixfolge ist. Im zweiten Schritt wird bewiesen, dass diese Matrixfolge im Sinne von $\boldsymbol{0} \preceq \boldsymbol{H}^{[l]} \preceq \boldsymbol{Y}$ nach oben beschränkt ist und zudem $\boldsymbol{0} \preceq \boldsymbol{H}^{[l]} \preceq \boldsymbol{H}^{[l+1]}$ gilt. Daraus folgt die Konvergenz der Matrixfolge. Im letzten Schritt wird gezeigt, dass die konvergierte Folge die Bellman-Gleichung erfüllt und das zugehörige Regelgesetz optimal ist.

Das nachfolgende Lemma stellt zunächst eine Erweiterung von [ATLAK07, Lemma 1] auf den Solltrajektorienregelungsfall dar und zeigt, dass die verwendete Value Iteration äquivalent zu einer Matrixfolge $\boldsymbol{H}^{[l]}$ ist.

Lemma 3.6

Sei $\boldsymbol{H}^{[0]} = \boldsymbol{0}$, $\boldsymbol{R}^\mathsf{T} = \boldsymbol{R} \succ \boldsymbol{0}$, $\boldsymbol{Q}^\mathsf{T} = \boldsymbol{Q} \succeq \boldsymbol{0}$ *und* $(\boldsymbol{A}, \boldsymbol{B})$ *steuerbar. Die durch* (3.112) *und* (3.117) *beschriebene Value Iteration ist äquivalent zur Iteration*

$$\boldsymbol{H}^{[l+1]} = \boldsymbol{G} + \gamma \boldsymbol{M}\left(-\boldsymbol{K}^{[l]}\right)^\mathsf{T} \boldsymbol{H}^{[l]} \boldsymbol{M}\left(-\boldsymbol{K}^{[l]}\right) \tag{3.119}$$

mit

$$\boldsymbol{G} := \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{0} & -\boldsymbol{Q} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R} & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{Q} & \boldsymbol{0} & \boldsymbol{Q} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{3.120}$$

und

$$\boldsymbol{M}\left(-\boldsymbol{K}^{[l]}\right) := \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ -\boldsymbol{K}_\mathrm{x}^{[l]}\boldsymbol{A} & -\boldsymbol{K}_\mathrm{x}^{[l]}\boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{K}_1^{[l]} & \cdots & -\boldsymbol{K}_{n_\mathrm{h}-1}^{[l]} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_n & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_n & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{I}_n \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \end{bmatrix}, \tag{3.121}$$

wobei $\boldsymbol{K}^{[l]} = \begin{bmatrix} \boldsymbol{K}_\mathrm{x}^{[l]} & \boldsymbol{K}_1^{[l]} & \cdots & \boldsymbol{K}_{n_\mathrm{h}}^{[l]} \end{bmatrix} =: \boldsymbol{K}\left(\boldsymbol{H}^{[l]}\right)$ *analog zu* (3.117) *aus* $\boldsymbol{H}^{[l]}$ *folgt.*

Beweis:
Mit quadratischen Basisfunktionen $\boldsymbol{\phi}(\cdot)$ (vgl. Lemma 3.4 und Lemma 3.5) ergibt sich

$$\hat{\boldsymbol{w}}^{[l]\mathsf{T}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{y}}_{k+1}^{*[l]}\right) = \frac{1}{2}\tilde{\boldsymbol{y}}_{k+1}^{*[l]\mathsf{T}}\boldsymbol{H}^{[l]}\tilde{\boldsymbol{y}}_{k+1}^{*[l]}. \tag{3.122}$$

Daraus folgt aufgrund von $\tilde{\boldsymbol{y}}_{k+1}^{*[l]} = \boldsymbol{M}\left(-\boldsymbol{K}^{[l]}\right)\tilde{\boldsymbol{y}}_k$

$$r_k + \gamma\hat{\boldsymbol{w}}^{[l]\mathsf{T}}\boldsymbol{\phi}\left(\tilde{\boldsymbol{y}}_{k+1}^{*[l]}\right) = \frac{1}{2}\tilde{\boldsymbol{y}}_k^{\mathsf{T}}\underbrace{\left(\boldsymbol{G} + \gamma\boldsymbol{M}\left(-\boldsymbol{K}^{[l]}\right)^{\mathsf{T}}\boldsymbol{H}^{[l]}\boldsymbol{M}\left(-\boldsymbol{K}^{[l]}\right)\right)}_{=\boldsymbol{H}^{[l+1]}}\tilde{\boldsymbol{y}}_k. \tag{3.123}$$

Aus (3.123), (3.113) und (3.114) folgt, dass (3.116) äquivalent zu

$$\begin{aligned}\hat{\boldsymbol{w}}^{[l+1]} &= \left(\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathsf{T}}\begin{bmatrix}\frac{1}{2}\tilde{\boldsymbol{y}}_{k-M+1}^{\mathsf{T}}\boldsymbol{H}^{[l+1]}\tilde{\boldsymbol{y}}_{k-M+1}\\ \vdots \\ \frac{1}{2}\tilde{\boldsymbol{y}}_k^{\mathsf{T}}\boldsymbol{H}^{[l+1]}\tilde{\boldsymbol{y}}_k\end{bmatrix}\\ &= \underbrace{\left(\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi}\right)^{-1}\left(\boldsymbol{\Phi}^{\mathsf{T}}\boldsymbol{\Phi}\right)}_{=\boldsymbol{I}}\operatorname{vecr}\left(\boldsymbol{H}^{[l+1]}\right)\end{aligned} \tag{3.124}$$

ist[66], wobei im letzten Schritt der Zusammenhang

$$\frac{1}{2}\tilde{\boldsymbol{y}}_\kappa^{\mathsf{T}}\boldsymbol{H}^{[l+1]}\tilde{\boldsymbol{y}}_\kappa = \boldsymbol{\phi}\left(\tilde{\boldsymbol{y}}_\kappa\right)^{\mathsf{T}}\operatorname{vecr}\left(\boldsymbol{H}^{[l+1]}\right), \tag{3.125}$$

$\kappa = k - M + 1, k - M + 2, \dots, k$, genutzt wird. Da $\boldsymbol{H}^{[l+1]}$ eine symmetrische Matrix ist, die aus den nicht-redundanten Elementen von $\hat{\boldsymbol{w}}^{[l+1]}$ gebildet wird, folgt aus (3.124) direkt, dass die Folge $\hat{\boldsymbol{w}}^{[l]}$, d. h. die Value Iteration, äquivalent zur Folge (3.119) ist. □

Das folgende technische Lemma wird für den Beweis von Lemma 3.8 benötigt.

Lemma 3.7
Sei $\boldsymbol{H}^{[0]} = \boldsymbol{0}$, $\boldsymbol{R}^{\mathsf{T}} = \boldsymbol{R} \succ \boldsymbol{0}$ und $\boldsymbol{Q}^{\mathsf{T}} = \boldsymbol{Q} \succeq \boldsymbol{0}$. Dann ist

$$\boldsymbol{u}_{k+\kappa} = -\boldsymbol{K}^{[l]}\begin{bmatrix}\boldsymbol{x}_{k+\kappa}^{\mathsf{T}} & \boldsymbol{x}_{\mathrm{r},k+\kappa+1}^{\mathsf{T}} & \dots & \boldsymbol{x}_{\mathrm{r},k+\kappa+n_{\mathrm{h}}}^{\mathsf{T}}\end{bmatrix}, \quad \forall l > 0, \tag{3.126}$$

wobei $\boldsymbol{K}^{[l]} = \boldsymbol{K}\left(\boldsymbol{H}^{[l]}\right)$ analog zu (3.117) aus $\boldsymbol{H}^{[l]}$ berechnet wird, die eindeutige Lösung, die

$$\hat{Q}_\kappa^{[l]} = \frac{1}{2}\tilde{\boldsymbol{y}}_{k+\kappa}^{\mathsf{T}}\boldsymbol{H}^{[l]}\tilde{\boldsymbol{y}}_{k+\kappa} \tag{3.127}$$

minimiert.

[66] Durch $\operatorname{vecr}(\cdot)$ wird die symmetrische Matrix $\boldsymbol{H}^{[l]}$ in einen Vektor überführt, d. h. es gilt $\operatorname{vecr}\left(\boldsymbol{H}^{[l]}\right) = \hat{\boldsymbol{w}}^{[l]}$.

Beweis:
Aufgrund von $\boldsymbol{H}^{[0]} = \boldsymbol{0} \succeq \boldsymbol{0}$, $\boldsymbol{R} \succ \boldsymbol{0}$ und $\boldsymbol{Q} \succeq \boldsymbol{0}$ folgt aus (3.119), dass $\boldsymbol{H}^{[l]} \succeq \boldsymbol{0}, \forall l > 0$. Weiterhin ist wegen $\boldsymbol{R} \succ \boldsymbol{0}$ ersichtlich, dass $\boldsymbol{h}_{\mathrm{uu}}^{[l]} \succ \boldsymbol{0}, \forall l > 0$, gilt. Weil aus $\frac{\partial \hat{Q}_{\kappa}^{[l]}}{\partial \boldsymbol{u}_{k+\kappa}} = \boldsymbol{0}$ die Reglermatrix $\boldsymbol{K}^{[l]}$ folgt und sich wegen $\boldsymbol{h}_{\mathrm{uu}}^{[l]} \succ \boldsymbol{0}$ zudem $\frac{\partial^2 \hat{Q}_{\kappa}^{[l]}}{\partial \boldsymbol{u}_{k+\kappa}^2} \succ \boldsymbol{0}$ ergibt, gilt die Aussage von Lemma 3.7. □

Im Folgenden werde der durch

$$F\left(\boldsymbol{\Omega}^{[l]}, \boldsymbol{\Gamma}^{[l]}\right) := \boldsymbol{G} + \gamma \boldsymbol{M}\left(\boldsymbol{\Gamma}^{[l]}\right)^{\mathsf{T}} \boldsymbol{\Omega}^{[l]} \boldsymbol{M}\left(\boldsymbol{\Gamma}^{[l]}\right) \tag{3.128}$$

definierte Operator verwendet. Somit gilt nach (3.119) $F\left(\boldsymbol{H}^{[l]}, -\boldsymbol{K}^{[l]}\right) = \boldsymbol{H}^{[l+1]}$.

Um zu beweisen, dass $\boldsymbol{H}^{[l]}$ nach Lemma 3.6 im Sinne von $\boldsymbol{0} \preceq \boldsymbol{H}^{[l]} \preceq \boldsymbol{Y}$ nach oben beschränkt ist, wird zunächst das nachfolgende technische Lemma benötigt, das eine Generalisierung von [Lan97, Lemma B.1.1] auf die solltrajektorienabhängige Q-Function darstellt. Die Kenntnis der analytischen Struktur von $\boldsymbol{H}$ (vgl. Satz 3.4) ist hierbei wesentlich für die Erweiterung auf den Solltrajektorienfall.

Lemma 3.8
Sei $\boldsymbol{W}^{[l]}$ eine beliebige Matrixfolge, wobei $\boldsymbol{W}^{[l]}$ dieselbe Dimension wie $\boldsymbol{K}^{[l]}$ besitze, zudem gelte $\boldsymbol{0} \preceq \boldsymbol{H}^{[0]} \preceq \boldsymbol{Z}^{[0]}$. Dann ergibt sich mit dem in (3.128) *definierten Operator $F(\cdot)$ für die Matrixfolgen*

$$\boldsymbol{Z}^{[l+1]} := F\left(\boldsymbol{Z}^{[l]}, \boldsymbol{W}^{[l]}\right) \tag{3.129}$$

und

$$\boldsymbol{H}^{[l+1]} = F\left(\boldsymbol{H}^{[l]}, -\boldsymbol{K}\left(\boldsymbol{H}^{[l]}\right)\right), \tag{3.130}$$

dass

$$\boldsymbol{0} \preceq \boldsymbol{H}^{[l+1]} \preceq \boldsymbol{Z}^{[l+1]} \tag{3.131}$$

gilt.

Beweis:
Der Beweis ist in Anhang A.2 gegeben. □

Im nächsten Schritt wird die Beschränktheit von $\boldsymbol{H}^{[l]}$ im Sinne von $\boldsymbol{0} \preceq \boldsymbol{H}^{[l]} \preceq \boldsymbol{Y}$ gezeigt. Für den Regulationsfall, d. h. für eine Q-Function ohne Solltrajektorienvorgabe, wurde diese Beschränktheit von Landelius [Lan97, Lemma B.1.2] bewiesen. Im Gegensatz dazu wird im

Folgenden der allgemeinere Solltrajektorienregelungsfall, d. h. die durch (3.119) beschriebene Iteration, betrachtet.

Lemma 3.9
Sei $(\boldsymbol{A}, \boldsymbol{B})$ *steuerbar und* $\boldsymbol{H}^{[l]}$ *die durch* (3.119) *beschriebene Folge mit* $\boldsymbol{H}^{[0]} = \mathbf{0}$. *Dann existiert eine Matrix* $\boldsymbol{Y}$, *sodass* $\mathbf{0} \preceq \boldsymbol{H}^{[l]} \preceq \boldsymbol{Y}$ *gilt.*

Beweis:
Siehe Anhang A.3. □

Nach dieser Vorarbeit kann schließlich das Hauptergebnis der Konvergenzanalyse formuliert werden. Dieses besagt, dass die Value Iteration gegen das optimale Gewicht $\boldsymbol{w}^*$ und somit gegen die optimale Reglermatrix $\boldsymbol{K}^*$ des Solltrajektorienfolgeregelungsproblems, das durch Problem 3.3 gegeben ist, konvergiert.

Satz 3.5
Sei $\boldsymbol{R}^\intercal = \boldsymbol{R} \succ \mathbf{0}$, $\boldsymbol{Q}^\intercal = \boldsymbol{Q} \succeq \mathbf{0}$, $(\boldsymbol{A}, \boldsymbol{B})$ *steuerbar und* $\hat{\boldsymbol{w}}^{[0]} = \mathbf{0}$, *d. h.* $\boldsymbol{H}^{[0]} = \mathbf{0}$. *Dann führt im Fall einer erfüllten Anregungsbedingung* (3.115) *die durch* (3.112) *und* (3.117) *gegebene Value Iteration zu* $\boldsymbol{H}^{[l]} \to \boldsymbol{H}$, *d. h. zur Konvergenz von* $\hat{\boldsymbol{w}}^{[l]} \to \boldsymbol{w}^*$ *und* $\boldsymbol{K}^{[l]} \to \boldsymbol{K}^*$.

Beweis:
Der Beweis erfolgt mittels vollständiger Induktion. Nach Lemma 3.6 ist die betrachtete Value Iteration äquivalent zur Folge $\boldsymbol{H}^{[l]}$ (vgl. (3.119)). Mit $\boldsymbol{Z}^{[0]} = \boldsymbol{H}^{[0]}$ und

$$\boldsymbol{Z}^{[l+1]} = F\Big(\boldsymbol{Z}^{[l]}, -\boldsymbol{K}\Big(\boldsymbol{H}^{[l+1]}\Big)\Big) \tag{3.132}$$

folgt

$$\mathbf{0} \preceq \boldsymbol{H}^{[l]} \preceq \boldsymbol{Z}^{[l]} \tag{3.133}$$

aus Lemma 3.8. Aufgrund von $\boldsymbol{H}^{[0]} = \mathbf{0}$ und mit $\boldsymbol{G}$ aus (3.120) folgt $\boldsymbol{H}^{[1]} = \boldsymbol{G} \succeq \mathbf{0}$ und somit der Induktionsanfang

$$\boldsymbol{H}^{[1]} - \boldsymbol{Z}^{[0]} \succeq \mathbf{0}. \tag{3.134}$$

Die Induktionsbehauptung sei durch

$$\boldsymbol{H}^{[l]} - \boldsymbol{Z}^{[l-1]} \succeq \mathbf{0} \tag{3.135}$$

gegeben. Es gilt

$$\boldsymbol{H}^{[l+1]} - \boldsymbol{Z}^{[l]} = \gamma \boldsymbol{M}\Big(-\boldsymbol{K}\Big(\boldsymbol{H}^{[l]}\Big)\Big)^\intercal \Big(\boldsymbol{H}^{[l]} - \boldsymbol{Z}^{[l-1]}\Big) \boldsymbol{M}\Big(-\boldsymbol{K}\Big(\boldsymbol{H}^{[l]}\Big)\Big) \succeq \mathbf{0}. \tag{3.136}$$

Daraus folgt

$$\mathbf{0} \preceq \boldsymbol{H}^{[l]} \preceq \boldsymbol{Z}^{[l]} \preceq \boldsymbol{H}^{[l+1]}. \tag{3.137}$$

Da die Matrixfolge $\boldsymbol{H}^{[l]}$ nach Lemma 3.9 durch $\boldsymbol{Y}$ nach oben beschränkt ist und wegen (3.137)

$$\mathbf{0} \preceq \boldsymbol{H}^{[l]} \preceq \boldsymbol{H}^{[l+1]} \tag{3.138}$$

gilt, existiert der Grenzwert $\boldsymbol{H}^{[\infty]}$, d. h. die Value Iteration konvergiert zu

$$\boldsymbol{H}^{[\infty]} = \boldsymbol{G} + \gamma \boldsymbol{M}\left(-\boldsymbol{K}\left(\boldsymbol{H}^{[\infty]}\right)\right)^{\mathsf{T}} \boldsymbol{H}^{[\infty]} \boldsymbol{M}\left(-\boldsymbol{K}\left(\boldsymbol{H}^{[\infty]}\right)\right). \tag{3.139}$$

Weiterhin minimiert $-\boldsymbol{K}\left(\boldsymbol{H}^{[\infty]}\right)$ nach Lemma 3.7 $\hat{Q}_0^{[\infty]}$. Daraus ergibt sich

$$\lim_{l \to \infty} \delta_k^{[l]} = r_k + \gamma \tilde{\boldsymbol{y}}_{k+1}^{*\mathsf{T}} \boldsymbol{H}^{[\infty]} \tilde{\boldsymbol{y}}_{k+1}^{*} - \tilde{\boldsymbol{y}}_k^{\mathsf{T}} F\left(\boldsymbol{H}^{[\infty]}, -\boldsymbol{K}\left(\boldsymbol{H}^{[\infty]}\right)\right) \tilde{\boldsymbol{y}}_k = 0. \tag{3.140}$$

Somit ist die Bellman-Gleichung für $\boldsymbol{H}^{[\infty]}$ erfüllt und es folgt $\boldsymbol{H}^{[l]} \to \boldsymbol{H}^{[\infty]} = \boldsymbol{H}$, d. h. $\hat{\boldsymbol{w}}^{[l]} \to \boldsymbol{w}^*$ und $\boldsymbol{K}^{[l]} \to \boldsymbol{K}^*$. □

3.3.4 Ergebnisse

Im Folgenden wird der vorgestellte ADP-basierte Solltrajektorienfolgeregler, der dem Sollzustandsverlauf auf einem gleitenden Vorausschauhorizont der Länge n_h optimal im Sinne von Problem 3.3 folgt, und der ohne explizite Kenntnis der Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ trainiert wird, simulativ ausgewertet. Die Ergebnisse werden zudem mit einer ADP-Methode aus der Literatur verglichen. Die Anzahl der Datentupel werde zunächst zu $M = 1{,}2h$ gesetzt, wobei h jeweils aus (3.107) folgt. Die Abbruchbedingung wird zu $e_{\hat{\boldsymbol{w}}} = 10^{-3}$ gewählt, die Diskontierung zunächst zu $\gamma = 0{,}9$ gesetzt und der Vorausschauhorizont, auf dem der Sollzustandsverlauf gegeben ist, zu $n_\mathrm{h} = 10$ gewählt. Der Anfangszustand des Systems sei jeweils $\boldsymbol{x}_0 = \mathbf{0}$.

3.3.4.1 Simulationsbeispiele

Betrachtet werden im Folgenden zwei Simulationsbeispiele. Das erste System ist ein rotatorisches Feder-Masse-Dämpfer-System zweiter Ordnung, das zweite System ein lineares Einspurmodell eines vorderradgelenkten Fahrzeugs der Ordnung sechs. Beide Systemmodelle resultieren aus einer Tustin-Approximation eines zeitkontinuierlichen Modells mit einer Abtastzeit von 1 s.

System 1 – Feder-Masse-Dämpfer-System
Die zeitdiskrete Zustandsraumrepräsentation des rotatorischen Feder-Masse-Dämpfer-Systems ist durch

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} 0{,}99 & 0{,}9 \\ -0{,}02 & 0{,}8 \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} 0{,}01 \\ 0{,}02 \end{bmatrix} u_k \tag{3.141}$$

gegeben. Die Stellgröße u_k stellt dabei ein Drehmoment dar, welches auf das System wirkt. Weiterhin seien

$$\boldsymbol{Q} = \mathrm{diag}(100, 0) \text{ und } \boldsymbol{R} = 1 \tag{3.142}$$

die Parameter des Gütefunktionals (3.82). Somit liegt der Fokus darauf, den Systemzustand x_1 einem Sollwinkel $x_{1,\mathrm{ref}} = \alpha_{\mathrm{ref}}$, der auf einem gleitenden Vorausschauhorizont der Länge $n_\mathrm{h} = 10$ gegeben ist, folgen zu lassen.

Für dieses Optimierungsproblem liegen für $\gamma = 0{,}9$ die Systempole des optimal geregelten Systems alle im Einheitskreis, da $(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_\mathrm{x}^*)$ einen doppelten Eigenwert bei $0{,}7503$ aufweist. Das hinreichende Stabilitätskriterium nach Satz 3.3 (vgl. Bemerkung 3.9) ist wegen $b_\gamma = 0{,}7067 < 0{,}9 = \gamma$ ebenfalls erfüllt, aus einer korrekt gelernten Q-Function lässt sich für dieses Szenario somit auch ohne explizite Kenntnis der Systemmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ auf Stabilität des geregelten Systems schließen.

Während des Trainingsvorgangs wird für dieses Simulationsbeispiel für die erforderliche Anregung (vgl. (3.115)) der additive Rauschterm $\boldsymbol{\xi}_k$ der Stellgröße in (3.118) mit einer Varianz von $\boldsymbol{\mathcal{V}} = 0{,}01$ gewählt. Außerdem wird während des Trainings mittelwertfreies weißes Gaußsches Rauschen $\xi_\mathrm{ref} \sim \mathcal{N}(0; 0{,}01)$ zum Solltrajektorienverlauf addiert.

System 2 – Lineares Einspurmodell
Das zweite betrachtete Beispielsystem ist ein lineares Einspurmodell sechster Ordnung (vgl. [Fla16, Anhang B]). Dieses ist durch

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} -0{,}633 & -0{,}035 & 0 & 0 & -1{,}4\cdot10^{-4} & -4{,}5\cdot10^{-3} \\ 2{,}445 & -0{,}771 & 0 & 0 & 5{,}1\cdot10^{-3} & 0{,}164 \\ 1{,}222 & 0{,}115 & 1 & 0 & 2{,}6\cdot10^{-3} & 0{,}082 \\ 15{,}890 & 0{,}794 & 20 & 1 & 0{,}024 & 0{,}774 \\ 6{,}197 & -0{,}323 & 0 & 0 & -0{,}925 & -1{,}587 \\ 3{,}099 & -0{,}161 & 0 & 0 & 0{,}038 & 0{,}206 \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} -2{,}7\cdot10^{-4} \\ 9{,}9\cdot10^{-3} \\ 4{,}9\cdot10^{-3} \\ 0{,}047 \\ 0{,}145 \\ 0{,}073 \end{bmatrix} u_k \tag{3.143}$$

gegeben. Die physikalische Bedeutung der Stellgröße u_k und des Systemzustands $\boldsymbol{x}_k$ ist in Anhang A.4 erläutert. Da im Folgenden die Abweichung von $x_{4,\mathrm{ref}} = y_\mathrm{ref}$ bestraft werden soll, werde

$$\boldsymbol{Q} = \mathrm{diag}(0, 0, 0, 100, 0, 0) \text{ und } \boldsymbol{R} = 1 \tag{3.144}$$

gewählt.

Auch bei diesem Beispiel liegen für $\gamma = 0{,}9$ die Systempole des optimal geregelten Systems im Inneren des Einheitskreises. Konkret ergeben sich für $(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_\mathrm{x}^*)$ doppelte Eigenwerte bei $0{,}4834$ und $0{,}7136$ sowie einfache Eigenwerte bei $0{,}0526$ und $0{,}8675$. Obwohl das optimal geregelte System somit auch für $\gamma = 0{,}9$ stabil ist, liefert das durch Satz 3.3 gegebene lediglich hinreichende Kriterium in diesem Fall wegen $b_\gamma \approx 1$ keine eindeutige Aussage. Daher soll für dieses Beispiel im Folgenden zusätzlich zum Fall $\gamma = 0{,}9$ noch der Fall $\gamma = 1$ betrachtet

werden. Für $\gamma = 1$ hat $(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_{\mathrm{x}}^{*})$ doppelte Eigenwerte bei $0{,}4556$ und $0{,}7136$ sowie einfache Eigenwerte bei $0{,}0507$ und $0{,}8675$.

Zur Anregung (vgl. (3.115)) wird der additive Rauschterm $\boldsymbol{\xi}_k$ mit $\boldsymbol{\mathcal{V}} = 0{,}03$ in (3.118) verwendet. Zudem wird während des Trainingsvorgangs mittelwertfreies weißes Gaußsches Rauschen $\xi_{\mathrm{ref}} \sim \mathcal{N}\left(0; 0{,}01\right)$ zum Solltrajektorienverlauf addiert.

3.3.4.2 Auswertungsmethodik

Im Folgenden bezeichne α_{ref} die Sollwinkeltrajektorie für Beispielsystem 1 und y_{ref} die Solltrajektorie für Beispielsystem 2. Die im Rahmen dieser Arbeit vorgestellte ADP-Solltrajektorienfolgeregelungsmethode, die den exakten Solltrajektorienverlauf auf einem gleitenden Vorausschauhorizont der Länge n_{h} in die Q-Function integriert, soll mit Literaturmethoden, die annehmen, der Solltrajektorienverlauf werde durch $\boldsymbol{f}_{\boldsymbol{x}_{\mathrm{r}}}(\boldsymbol{x}_{\mathrm{r},k}) = \boldsymbol{F}_{\mathrm{ref}}\boldsymbol{x}_{\mathrm{r},k}$ beschrieben, verglichen werden. Um diesen Vergleich zu ermöglichen, wird für die während des Trainingsvorgangs verwendeten Solltrajektorien

$$\alpha_{\mathrm{ref}} = y_{\mathrm{ref}} = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}_{\mathrm{r},k}, \tag{3.145}$$

mit

$$\boldsymbol{x}_{\mathrm{r},k+1} = \underbrace{\begin{bmatrix} 0{,}9801 & 0{,}1987 \\ -0{,}1987 & 0{,}9801 \end{bmatrix}}_{\boldsymbol{F}_{\mathrm{ref}}} \boldsymbol{x}_{\mathrm{r},k}, \tag{3.146}$$

angenommen. Zu Vergleichszwecken wird sowohl ein ADP-Regler entsprechend Algorithmus 3.2 und ein ADP-Regler entsprechend [LLHW16], [KLM+14], nachfolgend als Exosystem-Methode bezeichnet, trainiert. Der Exosystem-Ansatz wird folglich bezüglich des erweiterten Systems

$$\tilde{\boldsymbol{x}}_{k+1} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{F}_{\mathrm{ref}} \end{bmatrix} \tilde{\boldsymbol{x}}_k + \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{0} \end{bmatrix} u_k, \tag{3.147}$$

mit $\tilde{\boldsymbol{x}}_k := \begin{bmatrix} \boldsymbol{x}_k^\mathsf{T} & \boldsymbol{x}_{\mathrm{r},k}^\mathsf{T} \end{bmatrix}^\mathsf{T}$ trainiert.

Nach Abschluss des Trainingsprozesses wird der Solltrajektorienverlauf $\alpha_{\mathrm{ref}} = y_{\mathrm{ref}}$ deutlich variiert, um die Generalisierungsfähigkeit der ADP-Regler auf von (3.146) abweichende Solltrajektorienverläufe zu untersuchen. Die Bewertung der Ergebnisse geschieht dabei auf zwei Arten. Einerseits wird nach Beendigung des Trainingsvorgangs der Trajektorienverlauf, der sich unter Verwendung der gelernten ADP-Regler ergibt (bezeichnet durch α_{ADP} und y_{ADP} bzw. $\alpha_{\mathrm{ADP,Exosystem}}$ und $y_{\mathrm{ADP,Exosystem}}$), mit dem Trajektorienverlauf α_{opt} bzw. y_{opt}, der sich aus der optimalen analytischen Lösung nach Satz 3.4 ergibt, verglichen. Die mittlere quadratische Abweichung wird dabei mit α_{RMS} bzw. y_{RMS} bezeichnet. Andererseits werden die durch Algorithmus 3.2 gelernten Gewichte $\hat{\boldsymbol{w}}$ der in dieser Arbeit vorgestellten ADP-Methode

mit der optimalen Lösung $\boldsymbol{w}^*$, d. h. den zu Satz 3.4 und Lemma 3.5 korrespondierenden Gewichten, verglichen[67]. Zugunsten einer Vergleichbarkeit für unterschiedliche Wertebereiche von $\hat{\boldsymbol{w}}$ wird der Absolutfehler jedes Gewichts bezüglich des größten absoluten Gewichts $\max_j \left|\{\boldsymbol{w}^*\}_j\right|$ normiert. Der Mittelwert dieser normierten Absolutfehler über alle h Gewichte wird durch

$$e_{\mathrm{I}} = \frac{1}{h} \sum_{i=1}^{h} \frac{\left|\{\hat{\boldsymbol{w}}\}_i - \{\boldsymbol{w}^*\}_i\right|}{\max_j \left|\{\boldsymbol{w}^*\}_j\right|} \tag{3.148}$$

gegeben, der Maximalwert des normierten Absolutfehlers hingegen durch

$$e_{\mathrm{II}} = \max_{i \in \{1,\ldots,h\}} \frac{\left|\{\hat{\boldsymbol{w}}\}_i - \{\boldsymbol{w}^*\}_i\right|}{\max_j \left|\{\boldsymbol{w}^*\}_j\right|}. \tag{3.149}$$

3.3.4.3 Simulationsergebnisse

Die Trajektorienabweichungen α_{RMS} (System 1) und y_{RMS} (System 2) von der optimalen Lösung sowie die Schätzfehler e_{I} und e_{II} der Gewichte $\hat{\boldsymbol{w}}$ sind in Tabelle 3.3 gegeben[68].

Die resultierenden Trajektorienverläufe des optimalen Reglers sowie der beiden ADP-Ansätze sind in Abbildung 3.7 (Beispielsystem 1) und Abbildung 3.8 (Beispielsystem 2 mit $\gamma = 0{,}9$) gegeben. Für $\gamma = 1$ ergibt sich für System 2 ein nahezu identischer Verlauf zur Diskontierung $\gamma = 0{,}9$, dieser ist der Vollständigkeit halber in Anhang A.5 (Abbildung A.2) gegeben. Die vertikale, punkt-gestrichelte Linie gibt hierbei den Zeitschritt an, ab dem $k \geq M$ gilt und somit die Value Iteration nach Algorithmus 3.2 durchgeführt wird. Diese erreicht nach 42 (System 1), 43 (System 2, $\gamma = 0{,}9$), bzw. 67 (System 2, $\gamma = 1$) Iterationen die durch $e_{\hat{\boldsymbol{w}}}$ beschriebene Abbruchbedingung. Der Solltrajektorienverlauf α_{ref} bzw. y_{ref} ist in Grau dargestellt. Die schwarze, gestrichelte Linie zeigt den Verlauf der optimalen Lösungen α_{opt} bzw. y_{opt}. In Rot ist der Verlauf gezeigt, der sich für die in dieser Arbeit vorgestellte ADP-Methode ergibt, wohingegen in Blau die Ergebnisse der Vergleichsmethode gezeigt sind.

Die Gewichtsfehler e_{I} und e_{II} in Tabelle 3.3 lassen erkennen, dass die optimalen Q-Function-Gewichte $\hat{\boldsymbol{w}}$ bei beiden Systemen erfolgreich gelernt werden. Die Abnahme der Fehlermaße e_{I} und e_{II} während der Value Iteration nach Algorithmus 3.2 ist in Abbildung 3.9 (System 1), Abbildung 3.10 (System 2 mit $\gamma = 0{,}9$) und Abbildung A.3 (System 2 mit $\gamma = 1$) gezeigt.

[67] Es sei anzumerken, dass die zu Vergleichszwecken verwendete Exosystem-Methode nicht $\boldsymbol{w}^*$ aus Satz 3.4 und Lemma 3.5 lernt, sondern die optimalen Gewichte, welche mit (3.147) korrespondieren. Daher werden die nachfolgend vorgestellten Bewertungsmaße nur für die in dieser Arbeit vorgestellte ADP-Solltrajektorienregelungsmethode berechnet, um zu bewerten, ob ein erfolgreicher Trainingsvorgang vorliegt.

[68] An dieser Stelle sei anzumerken, dass die Exosystem-Methode, welche als Vergleichsalgorithmus verwendet wird, aufgrund der Wahl von (3.146) für $\gamma = 1$ ungeeignet ist, da der durch $\boldsymbol{F}_{\mathrm{ref}}$ beschriebene Solltrajektorienverlauf nicht abklingt, da $\boldsymbol{F}_{\mathrm{ref}}$ zwei Eigenwerte auf dem Einheitskreis der komplexen Ebene aufweist.

System 1

	ADP	ADP, Exosystem
α_{RMS}	$2{,}1 \cdot 10^{-3}$	1,55
e_{I}	$6{,}2 \cdot 10^{-5}$	–
e_{II}	$2{,}1 \cdot 10^{-3}$	–

System 2

	ADP ($\gamma = 0{,}9$)	ADP ($\gamma = 1$)	ADP, Exosystem ($\gamma = 0{,}9$)
y_{RMS}	$6{,}5 \cdot 10^{-5}$	$5{,}2 \cdot 10^{-4}$	1,00
e_{I}	$9{,}5 \cdot 10^{-6}$	$4{,}2 \cdot 10^{-5}$	–
e_{II}	$1{,}0 \cdot 10^{-3}$	$4{,}5 \cdot 10^{-3}$	–

Tabelle 3.3: Trajektorienfolgefehler und Gewichtsfehler der betrachteten Simulationsbeispiele.

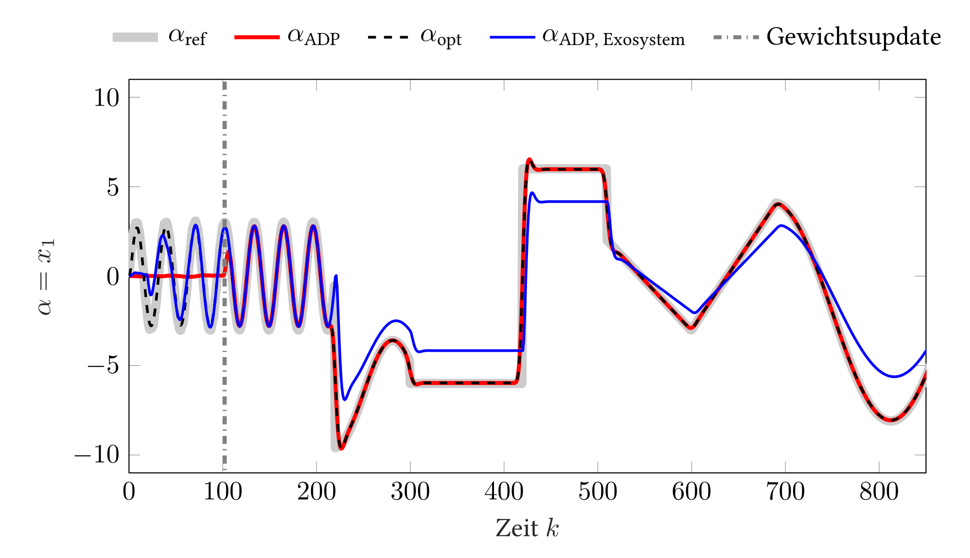


Abbildung 3.7: Ergebnis der ADP-Solltrajektorienregelung für System 1 (rotatorisches Feder-Masse-Dämpfer-System zweiter Ordnung).

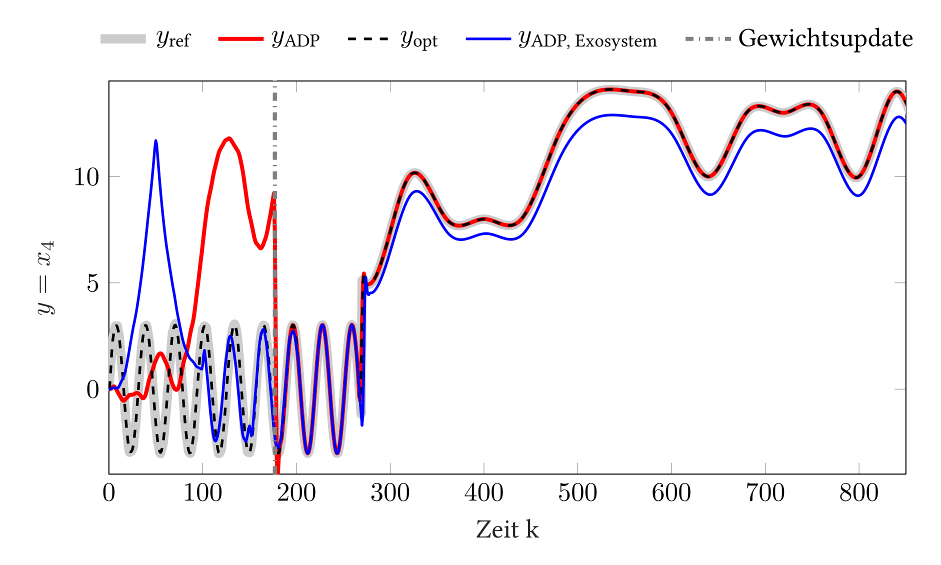


Abbildung 3.8: Ergebnis der ADP-Solltrajektorienregelung für System 2 (lineares Einspurmodell sechster Ordnung) für $\gamma = 0{,}9$.

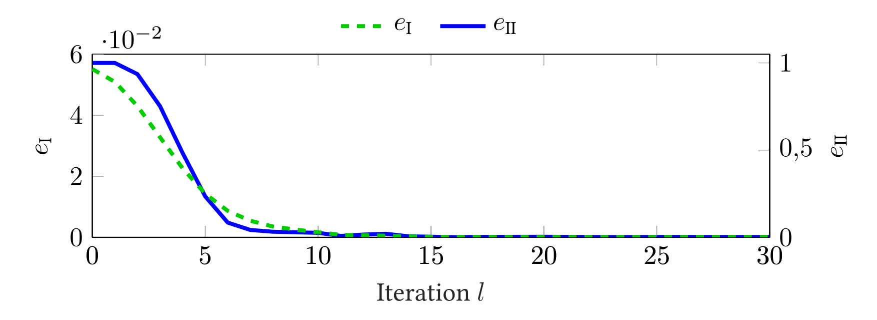


Abbildung 3.9: Gewichtsfehlerverlauf während der ersten 30 Iterationen des Lernvorgangs für System 1 (rotatorisches Feder-Masse-Dämpfer-System zweiter Ordnung). Hierbei stellt e_{I} (3.148) den Mittelwert und e_{II} (3.149) das Maximum des elementweisen absoluten Fehlers von $\hat{\boldsymbol{w}}$, jeweils durch $\max_j \left|\{\boldsymbol{w}^*\}_j\right|$ normiert, dar.

3.3.4.4 Einfluss von Messrauschen

Bisher wurde, wie in der ADP-Literatur üblich, angenommen, dass exakte Messungen des Systemzustands $\boldsymbol{x}_k$ vorliegen. Wenngleich der Fokus der vorliegenden Arbeit auf einer Formulierung einer Q-Function, die den Solltrajektorienverlauf einbezieht, und nicht auf robusten

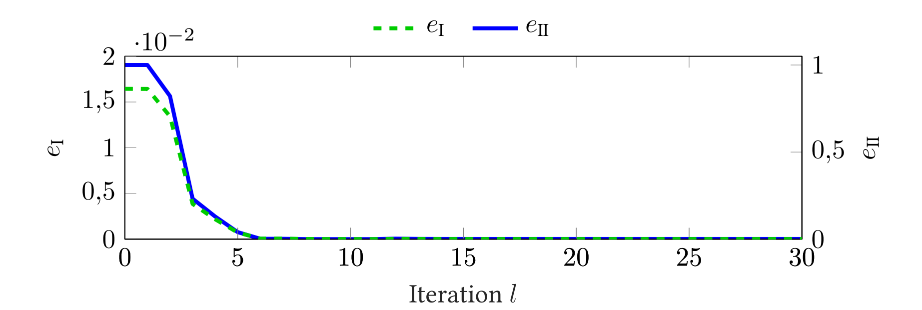


Abbildung 3.10: Gewichtsfehlerverlauf während des Lernvorgangs für System 2 (lineares Einspurmodell sechster Ordnung) für $\gamma = 0{,}9$. Hierbei stellt e_{I} (3.148) den Mittelwert und e_{II} (3.149) das Maximum des elementweisen absoluten Fehlers von $\hat{\boldsymbol{w}}$, jeweils durch $\max_j \left|\{\boldsymbol{w}^*\}_j\right|$ normiert, dar.

ADP-Methoden liegt[69], soll im Folgenden kurz der Einfluss unbekannten Messrauschens[70] $\boldsymbol{\nu}_k$ untersucht werden. Hierzu werde angenommen, dass dem ADP-basierten Regelungsansatz lediglich ein gemessener Systemzustand

$$\check{\boldsymbol{x}}_k = \boldsymbol{x}_k + \boldsymbol{\nu}_k \tag{3.150}$$

zur Verfügung steht, wobei $\boldsymbol{\nu}_k \in \mathbb{R}^n$ mittelwertfreies weißes Gaußsches Rauschen darstellt.

In dieser Situation werde der Regler für System 1 (3.141) mit $\gamma = 0{,}9$, $M = 10h$ (mit h wie in (3.107)) und für $n_{\mathrm{h}} = 10$ trainiert. Während des Trainings werden hierbei unterschiedliche Signal-Rausch-Verhältnisse $\mathrm{SNR}_{\mathrm{x}}(\boldsymbol{x}_k, \boldsymbol{\nu}_k)$ und $\mathrm{SNR}_{\mathrm{x_r}}(\boldsymbol{x}_{\mathrm{r},k}, \boldsymbol{\nu}_k)$ des Zustands $\boldsymbol{x}_k$ und Sollzustands $\boldsymbol{x}_{\mathrm{r},k}$ im Vergleich zum Messrauschen $\boldsymbol{\nu}_k$ betrachtet. Die Abweichung des gelernten Reglers $\boldsymbol{K}$ vom optimalen Regler $\boldsymbol{K}^*$ ist in Tabelle 3.4 gegeben. Die resultierenden Trajektorienverläufe sowie deren Standardabweichungen (std$(\cdot)$) für unterschiedliche Rauschamplituden sind in Abbildung 3.11 gezeigt. Je stärker das Anregungsrauschen im Vergleich zum Messrauschen dominiert, desto ähnlicher ist der gelernte Regler dem optimalen Regler und desto besser folgt der Winkel α der Solltrajektorie.

[69] Für erste Ansätze zu robusten ADP-Methoden, welche Gegenstand der aktuellen Forschung sind, sei auf [BJD10], [JJ14c], [NLW+19], [QZLY19], [WHL17], [WY18] und [ZCZL11] verwiesen.

[70] Dabei ist das unbekannte Messrauschen nicht mit dem Anregungsrauschen $\boldsymbol{\xi}_k$ in (3.118), welches bekannt ist und daher korrekt in den TD-Fehler einbezogen wird, zu verwechseln.

$\mathrm{SNR_x} = \mathrm{SNR_{x_r}}$	$\lVert \boldsymbol{K}^* - \boldsymbol{K} \rVert_2$	$\frac{\lVert \boldsymbol{K}^* - \boldsymbol{K} \rVert_2}{\lVert \boldsymbol{K}^* \rVert_2}$
10 dB	6,61	0,40
20 dB	1,61	$9{,}66 \cdot 10^{-2}$
30 dB	0,39	$2{,}37 \cdot 10^{-2}$
40 dB	0,15	$8{,}84 \cdot 10^{-3}$
50 dB	$4{,}75 \cdot 10^{-2}$	$2{,}84 \cdot 10^{-3}$
60 dB	$1{,}36 \cdot 10^{-2}$	$8{,}13 \cdot 10^{-4}$

Tabelle 3.4: Abweichungen der gelernten Regler vom optimalen Regler für unterschiedlich starkes Messrauschen. Die angegebenen Fehler stellen über 20 Trainingsdurchgänge gemittelte Werte dar.

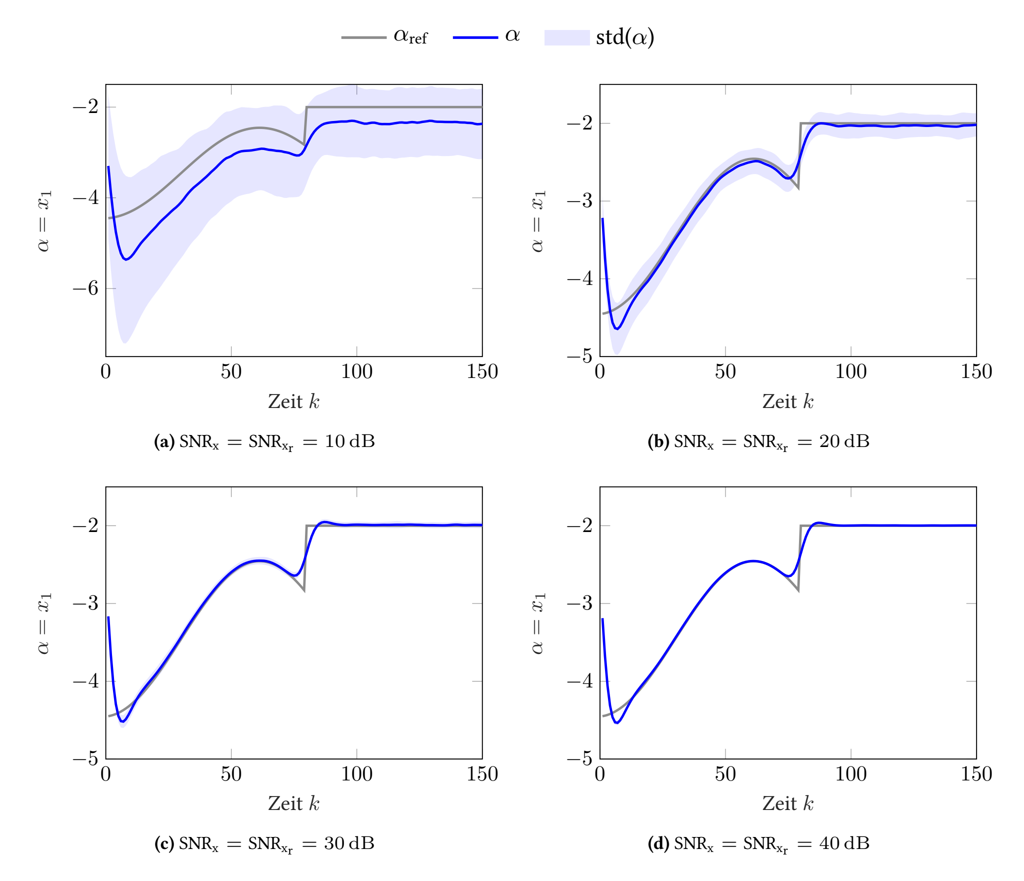


Abbildung 3.11: Trajektorienfolgeergebnisse für ADP-Regler, die bei unterschiedlich starkem Messrauschen trainiert und validiert wurden.

3.3.5 Diskussion der Simulationsergebnisse

Für den in dieser Arbeit vorgestellten ADP-Solltrajektorienfolgeregler, der den Sollzustandsverlauf auf einem gleitenden Vorausschauhorizont explizit einbezieht, wurde die Konvergenz gegen die optimale Lösung unter einer geeigneten Anregung nach (3.115) in Abschnitt 3.3.3.3 formal gezeigt. Insbesondere offenbaren Abbildung 3.9, Abbildung 3.10 und Abbildung A.3, dass die geschätzte solltrajektorienabhängige Q-Function $\hat{Q}_0$ für die untersuchten Simulationsbeispiele innerhalb weniger Iterationen gegen die optimale Q-Function Q_0 konvergiert. Aufgrund des optimalen Verhaltens entsprechend des durch (3.82) gegebenen Gütefunktionals und des Vorausschauhorizonts der Länge n_h weist der gelernte Regler des Weiteren prädiktives Verhalten auf und reagiert beispielsweise auf Sprünge bereits vor der Veränderung des aktuellen Sollzustands. Dies ist in Abbildung 3.12, welche die Detailansicht eines Sprungs aus Abbildung 3.7 zeigt, zu sehen.

Die Betrachtung von α_RMS (System 1) bzw. y_RMS (System 2) sowie die Abbildungen 3.7, 3.8 und 3.12 verdeutlichen zudem, dass der in dieser Arbeit präsentierte ADP-Solltrajektorienfolgeregelungsansatz erfolgreich und ohne weitere Modifikation auch Solltrajektorien, die von den während des Trainingsvorgangs verwendeten Trajektorienverläufen stark abweichen, folgen kann. Diese fundamentale Eigenschaft, welche die Verwendung flexibler und während des Trainingsvorgangs nicht vorliegender Solltrajektorien ermöglicht, ist insbesondere auf die explizite Abhängigkeit der gelernten Q-Function von einem beliebigen Solltrajektorienverlauf auf einem gleitenden Vorausschauhorizont der Länge n_h zurückzuführen. Im Gegensatz dazu sind existierende ADP-Methoden, die annehmen, der Sollzustandsverlauf folge der durch $\boldsymbol{f}_{\boldsymbol{x}_\mathrm{r}}$ beschriebenen Dynamik (beispielsweise [LLHW16] und [KLM+14]), zwar gut geeignet, solange diese Annahme erfüllt ist (wie dies beispielsweise in Abbildung 3.7 für $100 < k < 200$

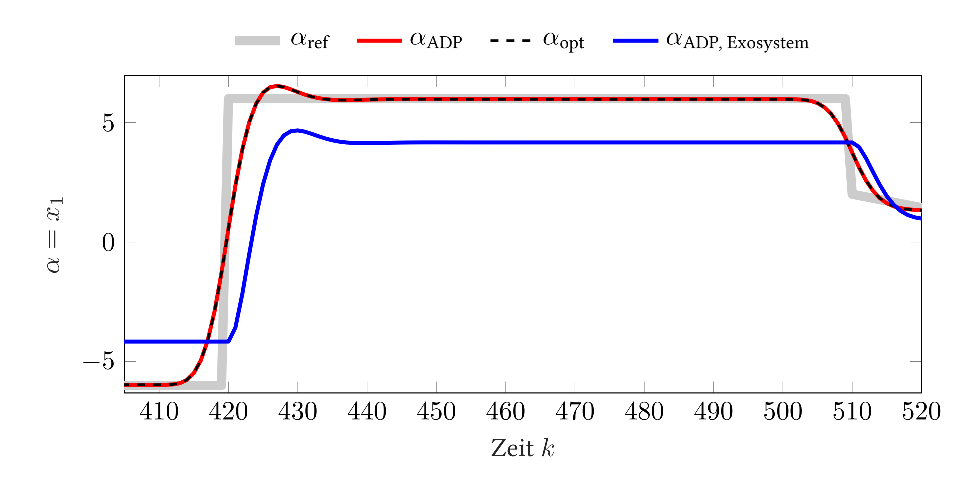


Abbildung 3.12: Detailansicht von Abbildung 3.7 (System 1), um das prädiktive Verhalten des vorgestellten ADP-Ansatzes aufgrund der Verwendung des gleitenden Vorausschauhorizonts der Länge n_h zu visualisieren.

der Fall ist), jedoch nimmt deren Güte ab, sobald der Solltrajektorienverlauf von (3.145)–(3.146) abweicht. Dieses Verhalten ist wenig verwunderlich, da bei diesen Reglern implizit die während des Trainingsvorgangs zugrunde liegende Exosystemdynamik auf die gelernte Q-Function Einfluss nimmt.

Besonders erwähnenswert ist auch, dass die genaue Kenntnis der Struktur der Matrix $\boldsymbol{H}_K$ als Ergebnis von Satz 3.4 und somit auch der Matrix $\boldsymbol{H}$ signifikant zu einer effizienten Wahl geeigneter Basisfunktionen beiträgt und eine wesentliche Reduktion der Anzahl h der zu lernenden Gewichte ermöglicht. Würde beispielsweise nur angenommen, $\boldsymbol{H}$ sei symmetrisch, so müssten bei den gezeigten Simulationsbeispielen $h = 325$ (System 1) bzw. $h = 2701$ (System 2) Gewichte geschätzt werden[71]. Unter Beachtung von Lemma 3.4 reduziert sich diese Anzahl um rund $24\,\%$ (System 1) bzw. $26\,\%$ (System 2). Nutzung der dünnbesetzten Matrix $\boldsymbol{Q}$ nach Lemma 3.5 reduziert die zu lernenden Gewichte schließlich sogar um etwa $74\,\%$ (System 1) bzw. $95\,\%$ (System 2) im Vergleich zur reinen Berücksichtigung der Symmetrie von $\boldsymbol{H}$ (vgl. Tabelle 3.2 für den Fall $n_\mathrm{h} = 10$). Wie (3.107) zeigt, beeinflusst die Wahl der Länge n_h des Vorausschauhorizonts direkt die Anzahl h der zu lernenden Gewichte. Für den Fall, dass der Solltrajektorienverlauf über einen großen Vorausschauhorizont bekannt ist, kann somit zwar ein potenziell besseres prädiktives Verhalten erreicht werden, die Komplexität des ADP-Problems erhöht sich jedoch entsprechend. Somit ist bei der Wahl einer geeigneten Vorausschauhorizontlänge n_h ein von der spezifischen Anwendung abhängiger Kompromiss einzugehen.

3.4 Resümee zur zeitdiskreten ADP-basierten Solltrajektorienfolgeregelung

Zusammenfassend wurde in Kapitel 3 erstmalig formal der Begriff zeitdiskreter ADP-kompatibler Solltrajektoriendarstellungen definiert. Zudem wurden zwei neuartige, flexible und effiziente ADP-basierte Solltrajektorienfolgeregelungsansätze präsentiert und analysiert. Ohne exakte Kenntnis der Systemdynamik können so Regelgesetze erlernt werden, die generalisieren, indem während des Trainingsvorgangs nicht gesehenen Solltrajektorienverläufen optimal gefolgt werden kann. Dies stellt einen wesentlichen Vorteil gegenüber bestehenden Vergleichsmethoden dar. Des Weiteren wird durch die vorgestellten Ansätze auch der zukünftige Verlauf der Solltrajektorie berücksichtigt, wodurch die resultierenden Regler prädiktives Verhalten aufweisen.

Die beiden neuen, zeitdiskreten Solltrajektoriendarstellungen sind dabei ADP-kompatibel nach Definition 3.1 (vgl. Bemerkung 3.2 und Bemerkung 3.7). Bei der in Abschnitt 3.2 präsentierten lokalen Beschreibung des Solltrajektorienverlaufs durch den Parameter $\boldsymbol{Z}_k$ und Basisfunktionsvektor $\boldsymbol{\rho}$ nach (3.15) entscheidet insbesondere die Wahl von $\boldsymbol{\rho}$ über die Approximationsfähigkeit des Sollzustandsverlaufs und die Anzahl h der zu lernenden Gewichte. Wenngleich eine lokale Approximation des Solltrajektorienverlaufs, wie beispielsweise die in

[71] Selbst in diesem Fall wird schon das Wissen genutzt, dass die Q-Function quadratisch ist.

Abschnitt 3.2.5.1 gezeigten kubischen Polynome, gewisse Einschränkungen an die Vorausschaufähigkeit von Solltrajektorienreglern implizieren kann, so bietet eine an die Problemstellung angepasste Wahl der Solltrajektoriendarstellung im aktuellen Zeitschritt k einen entscheidenden Kompromiss zwischen Flexibilität und einer möglichst kompakten Darstellung mit handhabbarer Dimension h des Gewichtsvektors $\hat{\boldsymbol{w}}$. Dieser neuartige Ansatz erlaubt somit erstmals allgemein die Entwicklung vorausschauender, flexibler, kompakter, modellfreier ADP-basierter Solltrajektorienfolgeregler.

Zudem wird in Abschnitt 3.3 der Fall betrachtet, direkt die Sollzustände, die auf einem gleitenden Vorausschauhorizont der Länge n_h gegeben sind, in die Q-Function zu integrieren, anstatt eine parametrische Beschreibung des Solltrajektorienverlaufs zu verwenden. Dieser Ansatz erlaubt einerseits eine völlig beliebige und exakte Vorgabe von Sollzuständen auf dem gleitenden Vorausschauhorizont, korreliert andererseits jedoch direkt mit einer quadratischen Zunahme der Dimension h des Gewichtsvektors $\hat{\boldsymbol{w}}$ bei steigender Vorausschauhorizontlänge n_h (vgl. beispielsweise (3.107)). Bei einer sehr großen Vorausschauhorizontlänge n_h stößt dieser Ansatz aufgrund der damit verbundenen unverhältnismäßigen Anzahl zu schätzender Critic-Gewichte somit an seine Grenzen. Beispielsweise zeigt Tabelle 3.2, dass für den Fall $n_h = 100$ selbst bei einem System zweiter Ordnung bereits Tausende Gewichte gelernt werden müssten. Dies geht neben einer höheren Anzahl benötigter Messdaten mit einer erhöhten Rechenzeit, einer potenziell erschwerten Erfüllung der Anregungsbedingung (3.115) und möglichen numerischen Problemen einher. Letztlich ist im Einzelfall abzuwägen, ob die direkte Verwendung der Sollzustände wie in Abschnitt 3.3 oder eine parametrische Approximation wie in Abschnitt 3.2 für eine konkrete Problemstellung von Vorteil ist. Aus theoretischer Sicht sei der Vollständigkeit halber noch angemerkt, dass die exakte Verwendung des Solltrajektorienverlaufs auf einem gleitenden Vorausschauhorizont der Länge n_h nach Abschnitt 3.3 letztlich als Spezialfall der parametrischen Darstellung aus Abschnitt 3.2 interpretiert werden kann. Hierbei entspricht $\boldsymbol{Z}_k$ den unveränderten Sollzuständen auf dem Vorausschauhorizont der Länge n_h und $\boldsymbol{\rho}(\kappa)$ fungiert als Schieberegister.

Schließlich sei noch zu betonen, dass die im vorliegenden Kapitel vorgestellten Methoden zur ADP-kompatiblen Solltrajektorienrepräsentation grundsätzlich unabhängig von der konkreten Wahl des ADP-Algorithmus sind. Während in Abschnitt 3.2 beispielhaft eine Policy Iteration und in Abschnitt 3.3 eine Value Iteration verwendet wurde, können die vorgestellten ADP-kompatiblen Solltrajektoriendarstellungen ebenso im Rahmen von Actor-Critic-Ansätzen angewandt werden. Hierzu sei exemplarisch auf das Anwendungsbeispiel in Abschnitt 6.1 verwiesen. Ein weiterer Vorteil der vorgestellten ADP-kompatiblen Solltrajektoriendarstellungen ist zudem, dass, wann immer bei den vorgestellten Ansätzen eine Off-Policy-Methode zum Einsatz kommt, vom System aufgezeichnete Daten $\boldsymbol{x}_k$, $\boldsymbol{u}_k$, $\boldsymbol{x}_{k+1}$ während des Trainingsvorgangs wiederverwendet werden können, um die Dateneffizienz zu steigern (vgl. Abschnitt 2.1.4.4).

Die vorgestellten, neuartigen, flexiblen Mechanismen für ADP-basierte Solltrajektorienfolgeregler beantworten somit für den Fall zeitdiskreter Problemstellungen die in Abschnitt 2.4.1 formulierte Forschungsfrage 1 nach geeigneten Solltrajektoriendarstellungen und ihrer effizienten Integration in modellfreie ADP-Ansätze. Im nächsten Kapitel wird, auf den Erkenntnissen des

vorliegenden Kapitels aufbauend, der zeitkontinuierliche ADP-basierte Solltrajektorienfolgeregelungsfall betrachtet.

4 Zeitkontinuierliche ADP-basierte Solltrajektorienfolgeregelung

Nachdem im vorigen Kapitel zeitdiskrete ADP-kompatible Solltrajektoriendarstellungen betrachtet wurden, wird in diesem Kapitel die in Abschnitt 2.4.1 formulierte Forschungsfrage 1 für den zeitkontinuierlichen Fall beantwortet. Zunächst wird die Definition ADP-kompatibler Sollzustandsverläufe (vgl. Abschnitt 3.1) auf den zeitkontinuierlichen Fall übertragen. Anschließend wird eine ADP-kompatible zeitkontinuierliche Solltrajektoriendarstellung[72] präsentiert, bei welcher der Solltrajektorienverlauf mithilfe eines von außen vorgebbaren Parameters definiert wird. Durch die explizite Abhängigkeit der Value Function nicht nur vom Systemzustand, sondern auch von diesem Trajektorienparameter, resultiert eine hohe Flexibilität der möglichen Solltrajektorienverläufe. Darauf aufbauend wird für linear-quadratische zeitkontinuierliche optimale Trajektorienfolgeregelungsprobleme der Einfluss eines globalen Diskontierungsfaktors untersucht sowie eine Problemstellung mit teilweiser Dämpfung formuliert. Für diese Problemstellung lassen sich anschließend Aussagen über die Stabilität der optimalen Lösung treffen. Die vorgestellte Solltrajektoriendarstellung wird in einen bestehenden ADP-Ansatz integriert, sodass der resultierende optimale Trajektorienfolgeregler ohne Kenntnis der Systemdynamik aus aufgezeichneten Daten erlernt werden kann. Simulationsergebnisse für verschiedene Hyperparameter der neuartigen ADP-kompatiblen Solltrajektoriendarstellung sowie eine anschließende Diskussion schließen das vorliegende Kapitel ab.

4.1 Definition ADP-kompatibler zeitkontinuierlicher Trajektorien

Analog zu Abschnitt 3.1 lässt sich auch für den zeitkontinuierlichen Fall eine ADP-kompatible Solltrajektoriendarstellung formulieren.

[72] Dieser Abschnitt basiert auf einem im Rahmen der vorliegenden Dissertation entstandenen Konferenzbeitrag [BKIH21].

Definition 4.1 (Zeitkontinuierliche ADP-kompatible Solltrajektoriendarstellung)
Eine mit dem ADP-Formalismus kompatible Solltrajektoriendarstellung mit dem Sollzustand $\boldsymbol{x}_\mathrm{r}(t) \in \mathcal{X}$ ist durch zeitinvariante Funktionen $\boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}} : \mathcal{Z} \to \mathcal{X}$ und $\boldsymbol{f}_{\boldsymbol{\zeta}} : \mathcal{Z} \to \mathbb{R}^{n_\zeta}$ mit

$$\boldsymbol{x}_\mathrm{r}(t) = \boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}(\boldsymbol{\zeta}(t)) \tag{4.1}$$

und

$$\dot{\boldsymbol{\zeta}}(t) = \boldsymbol{f}_{\boldsymbol{\zeta}}(\boldsymbol{\zeta}(t)) \tag{4.2}$$

charakterisiert, wobei $\boldsymbol{\zeta} \in \mathcal{Z} \subset \mathbb{R}^{n_\zeta}$, $\forall t$, $n_\zeta < \infty$. Zudem sei $\boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}(\cdot)$ stetig und $\boldsymbol{f}_{\boldsymbol{\zeta}}(\cdot)$ Lipschitz-stetig auf $\mathcal{Z}$.

Basierend auf Definition 4.1 lässt sich die folgende formale Aussage treffen.

Proposition 4.1
Die Solltrajektoriendarstellung sei nach Definition 4.1 ADP-kompatibel. Für ein durch die zeitkontinuierliche Systemdynamik (2.10) *beschriebenes System*[73] *und ein Lipschitz-stetiges Regelgesetz $\boldsymbol{\mu}(\cdot)$ gilt: Sind die durch*

$$\int_t^\infty r\Big(\boldsymbol{x}(\tau), \boldsymbol{x}_\mathrm{r}(\tau), \boldsymbol{\mu}\big(\boldsymbol{x}(\tau), \boldsymbol{x}_\mathrm{r}(\tau)\big)\Big)\,\mathrm{d}\tau \tag{4.3}$$

gegebenen Gesamtkosten (vgl. (2.20)*), die von der Solltrajektorie $\boldsymbol{x}_\mathrm{r}(\tau)$, $\tau \geq t$, nach Definition 4.1 abhängen, endlich*[74]*, dann können sie durch eine Value Function der Form $V^{\boldsymbol{\mu}}(\boldsymbol{x}(t), \boldsymbol{\zeta}(t))$ beschrieben werden.*

Beweis:
Sei

$$\begin{aligned} V^{\boldsymbol{\mu}}\big(\boldsymbol{x}(t), \boldsymbol{\zeta}(t)\big) &:= \int_t^\infty r\Big(\boldsymbol{x}(\tau), \boldsymbol{x}_\mathrm{r}(\tau), \boldsymbol{\mu}\big(\boldsymbol{x}(\tau), \boldsymbol{x}_\mathrm{r}(\tau)\big)\Big)\,\mathrm{d}\tau && \text{(4.4a)}\\ &= \int_t^\infty r\Big(\boldsymbol{x}(\tau), \boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}\big(\boldsymbol{\zeta}(\tau)\big), \boldsymbol{\mu}\Big(\boldsymbol{x}(\tau), \boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}\big(\boldsymbol{\zeta}(\tau)\big)\Big)\Big)\,\mathrm{d}\tau. && \text{(4.4b)} \end{aligned}$$

Aufgrund von $\dot{\boldsymbol{\zeta}}(\tau) = \boldsymbol{f}_{\boldsymbol{\zeta}}(\boldsymbol{\zeta}(\tau))$ mit $\boldsymbol{f}_{\boldsymbol{\zeta}}(\cdot)$ Lipschitz-stetig existiert ausgehend von $\boldsymbol{\zeta}(t)$ nach dem Satz von Picard-Lindelöf [BSMM08, S. 673] $\forall \tau \geq t$ eine eindeutige Lösung für $\boldsymbol{\zeta}(\tau) \in \mathcal{Z}$. Somit legen $\boldsymbol{\zeta}(t)$ und $\boldsymbol{f}_{\boldsymbol{\zeta}}(\cdot)$ $\forall \tau \geq t$ den Referenzparameter $\boldsymbol{\zeta}(\tau)$ in (4.4b) eindeutig fest.

[73] Ebenso gilt die Aussage von Proposition 4.1 auch für allgemeinere nichtlineare Systeme der Form $\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(\boldsymbol{x}(t), \boldsymbol{u}(t))$, aus Gründen der Einheitlichkeit wurde hier jedoch der eingangsaffine Fall formuliert.

[74] Für sinnvoll gestellte Probleme schließt dies neben der Stabilisierbarkeit des Systems und einem zulässigen Regelgesetz $\boldsymbol{\mu}$ auch ein, dass durch die Wahl des Solltrajektorienverlaufs oder im Falle einer Erweiterung auf ein durch γ diskontiertes Gütefunktional die Gesamtkosten endlich bleiben.

Analoges gilt ausgehend von $\boldsymbol{x}(t)$ und $\boldsymbol{\zeta}(t)$ für den Systemzustand $\boldsymbol{x}(\tau)$, $\forall \tau \geq t$, für den durch

$$\dot{\boldsymbol{x}}(\tau) = \boldsymbol{f}\big(\boldsymbol{x}(\tau)\big) + \boldsymbol{g}\big(\boldsymbol{x}(\tau)\big)\boldsymbol{\mu}\Big(\boldsymbol{x}(\tau), \boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}\big(\boldsymbol{\zeta}(\tau)\big)\Big) \tag{4.5}$$

geschlossenen Regelkreis. Daraus folgt, dass eine explizite Abhängigkeit der Value Function (4.4) von $\boldsymbol{x}(t)$ und $\boldsymbol{\zeta}(t)$ zur Beschreibung der Gesamtkosten geeignet ist. □

Eine ADP-kompatible Solltrajektoriendarstellung stellt somit sicher, dass die Gesamtkostenrepräsentation in Form einer Value Function lediglich eine explizite Abhängigkeit von $\boldsymbol{x}(t)$ und $\boldsymbol{\zeta}(t)$ aufweist. Die in den Abschnitten 2.2.2 und 2.2.3 diskutierten zeitkontinuierlichen ADP-Trajektorienfolgeregelungsmethoden erfüllen zwar die Kompatibilitätsforderung nach Definition 4.1[75], ermöglichen jedoch keine externe Vorgabe eines Solltrajektorienverlaufs. Nachfolgend wird eine nach Definition 4.1 kompatible Solltrajektoriendarstellung vorgestellt, die zudem den zukünftigen Referenzzustandsverlauf berücksichtigt.

4.2 Zeitkontinuierliche ADP-kompatible parametrierte Referenztrajektorie

Im nächsten Unterabschnitt wird zunächst eine ADP-kompatible Solltrajektoriendarstellung präsentiert. Der Einfluss eines global diskontierten Gütemaßes auf den resultierenden Optimalregler wird anschließend in Abschnitt 4.2.2 untersucht. Da es hierbei potenziell zu Instabilität des geschlossenen Regelkreises kommen kann, wird in Abschnitt 4.2.3 ein Optimierungsproblem mit gedämpfter Referenzdynamik als Alternative zur Verwendung global diskontierter Gütemaße vorgestellt und analysiert. Für dieses Optimierungsproblem wird in Abschnitt 4.2.4 ein ADP-basierter Regelungsansatz entworfen. Schließlich werden in Abschnitt 4.2.5 Simulationsergebnisse gezeigt, welche die Vorzüge der neuartigen ADP-kompatiblen Solltrajektoriendarstellung verdeutlichen.

4.2.1 Solltrajektoriendarstellung

Betrachtet werde die Sollzustandstrajektorie

$$\boldsymbol{x}_\mathrm{r}(t) = \begin{bmatrix} x_{\mathrm{r},1}(t) & x_{\mathrm{r},2}(t) & \dots & x_{\mathrm{r},n}(t) \end{bmatrix}^\mathsf{T}. \tag{4.6}$$

Da auch in diesem Abschnitt zugunsten der Analysierbarkeit der LQ-Fall betrachtet werden soll, wird für die Elemente $x_{\mathrm{r},i}(t)$, $\forall i = 1, \dots, n$, eine lineare Dynamik angesetzt.

75 Bei Methoden, bei denen ein stationärer Sollzustand vorgegeben wird, stellt $\boldsymbol{f}_{\boldsymbol{x}_\mathrm{r},\boldsymbol{\zeta}}(\cdot)$ eine Identitätsabbildung dar und es gilt $\boldsymbol{f}_{\boldsymbol{\zeta}}(\cdot) = \boldsymbol{0}$.

Konkret wird der Ansatz

$$x_{\mathrm{r},i}(t) = \sum_{j=1}^{J_i} \left(\sum_{k=0}^{\nu_{ij}-1} c_{ijk} t^k \right) e^{\lambda_{ij} t} = \sum_{j=1}^{J_i} x_{\mathrm{r},ij}(t) \tag{4.7}$$

mit den Parametern $c_{ijk} \in \mathbb{C}$ und den Hyperparametern $\lambda_{ij} \in \mathbb{C}$ und $\nu_{ij} \in \mathbb{N}$ gewählt. Hierbei bezeichnet J_i die Anzahl verschiedener Eigenwerte und ν_{ij} die algebraische Vielfachheit des Eigenwerts λ_{ij}. Die Parameter $c_{ijk} \in \mathbb{C}$ sind zunächst frei wählbar. Sie bestimmen den konkreten Verlauf der Sollzustandstrajektorie $\boldsymbol{x}_\mathrm{r}(t)$. Somit ermöglichen die Parameter c_{ijk} die direkte Beeinflussung des Solltrajektorienverlaufs von außen (vgl. Abbildung 2.4), im Gegensatz zu der in der Literatur häufig verwendeten globalen Referenztrajektorie ohne äußere Beeinflussungsmöglichkeit nach Abbildung 2.2 und Tabelle 2.2. Verglichen mit den in Abschnitt 2.2.3 erwähnten Methoden, die lediglich den aktuellen Sollzustand oder die aktuelle Abweichung von diesem verwenden (vgl. Abbildung 2.3), wird durch den hier vorgestellten neuen Ansatz außerdem der Verlauf der Solltrajektorie berücksichtigt. Es sei weiterhin erwähnt, dass die aus der Literatur bekannte stationäre Sollzustandsvorgabe einen Spezialfall des neu vorgestellten Ansatzes darstellt (vgl. Abschnitt 4.2.5.2). Der auf (4.6) und (4.7) basierende Ansatz wird nachfolgend analysiert und dessen ADP-Kompatibilität im Sinne von Definition 4.1 gezeigt.

Die Darstellung nach (4.7) entspricht der Summation von J_i Lösungen $x_{\mathrm{r},ij}$ homogener linearer Differenzialgleichungen[76]

$$\underbrace{a_{ij\nu_{ij}}}_{:=1} x_{\mathrm{r},ij}^{(\nu_{ij})}(t) + \dots + a_{ij1}\dot{x}_{\mathrm{r},ij} + a_{ij0}x_{\mathrm{r},ij} = 0 \tag{4.8}$$

mit ν_{ij}-fachem Eigenwert λ_{ij} [MBK19, Kapitel 4.7], deren charakteristische Polynome durch

$$p_{ij}(\lambda) = (\lambda - \lambda_{ij})^{\nu_{ij}} = \sum_{k=0}^{\nu_{ij}} \underbrace{\binom{\nu_{ij}}{k} (-\lambda_{ij})^{\nu_{ij}-k}}_{=:a_{ijk}} \lambda^k \tag{4.9}$$

(vgl. Binomischer Lehrsatz [AE10, Satz 8.4]) gegeben sind. Während die Parameter c_{ijk}, die den Solltrajektorienverlauf beschreiben, zur Laufzeit variiert werden können, werden die Hyperparameter λ_{ij} und ν_{ij} vorab festgelegt[77].

Unter der folgenden Annahme ist die Reellwertigkeit des Solltrajektorienverlaufs $\boldsymbol{x}_\mathrm{r}(t)$ gewährleistet.

[76] Die Notation $x_{\mathrm{r},ij}^{(k)}(t)$ bezeichnet die k-te zeitliche Ableitung von $x_{\mathrm{r},ij}(t)$.

[77] Wie sich später in Bemerkung 4.3 zeigen wird, stellt dies ADP-Kompatibilität gemäß Definition 4.1 sicher.

Annahme 4.1

Für alle reellwertigen Eigenwerte $\lambda_{ij} \in \mathbb{R}$ *sei* $c_{ijk} \in \mathbb{R}\ \forall k \in \{0, \ldots, \nu_{ij} - 1\}$. *Zudem gelte für jeden komplexwertigen Eigenwert* $\lambda_{ij} \in \mathbb{C}$ *und den dazu konjugiert-komplexen Eigenwert* $\lambda_{ij}^{\star}$, *gekennzeichnet durch den Index* $j^{\star}$ *(d. h.* $\lambda_{ij}^{\star} =: \lambda_{ij^{\star}}$*),* $\nu_{ij^{\star}} = \nu_{ij}$ *und* $c_{ij^{\star}k} = c_{ijk}^{\star}$.

Lemma 4.1

Unter Annahme 4.1 folgt für Sollzustandstrajektorien $\boldsymbol{x}_{\mathrm{r}}(t)$ *nach* (4.6) *und* (4.7) *Reellwertigkeit, d. h.* $\boldsymbol{x}_{\mathrm{r}}(t) \in \mathbb{R}^n$.

Beweis:

Siehe Anhang B.1. □

Bemerkung 4.1

Für mehrfache Eigenwerte λ_{ij} *bei null entspricht der Sollzustandsverlauf* $\boldsymbol{x}_{\mathrm{r}}(t)$ *einer Polynomfunktion und reelle Eigenwerte ungleich null resultieren in Exponentialfunktionen. Aus* (4.7) *folgt für reelle Eigenwerte* λ_{ij} *mit* $c_{ijk} \in \mathbb{R}$ *(*$\forall k \in \{0, \ldots, \nu_{ij} - 1\}$*) nach Annahme 4.1 für jedes der* J_i *Summenglieder*

$$x_{\mathrm{r},ij}(t) = \left(c_{ij0} + c_{ij1}t + \cdots + c_{ij(\nu_{ij}-1)}t^{(\nu_{ij}-1)}\right)\mathrm{e}^{\lambda_{ij}t}. \tag{4.10}$$

Konjugiert-komplexe Polpaare nach Annahme 4.1 führen zu harmonischen Funktionen, deren Amplitude und Phase durch $c_{ij0} \in \mathbb{C}$ *eingestellt werden können, da für* $\lambda_{ij} = \lambda_{\mathrm{R}} + \mathrm{j}\lambda_{\mathrm{I}} = \lambda_{ij^{\star}}^{\star}$ *und* $c_{ij0} = c_{\mathrm{R}} + \mathrm{j}c_{\mathrm{I}} = c_{ij^{\star}}^{\star}$ *gilt:*

$$\begin{aligned} x_{\mathrm{r},ij}(t) + x_{\mathrm{r},ij^{\star}}(t) &= (c_{\mathrm{R}} + \mathrm{j}c_{\mathrm{I}})\,\mathrm{e}^{(\lambda_{\mathrm{R}}+\mathrm{j}\lambda_{\mathrm{I}})t} + (c_{\mathrm{R}} - \mathrm{j}c_{\mathrm{I}})\,\mathrm{e}^{(\lambda_{\mathrm{R}}-\mathrm{j}\lambda_{\mathrm{I}})t} \\ &= c_{\mathrm{R}}\mathrm{e}^{\lambda_{\mathrm{R}}t}\left(\mathrm{e}^{\mathrm{j}\lambda_{\mathrm{I}}t} + \mathrm{e}^{-\mathrm{j}\lambda_{\mathrm{I}}t}\right) + \mathrm{j}c_{\mathrm{I}}\mathrm{e}^{\lambda_{\mathrm{R}}t}\left(\mathrm{e}^{\mathrm{j}\lambda_{\mathrm{I}}t} - \mathrm{e}^{-\mathrm{j}\lambda_{\mathrm{I}}t}\right) \\ &= 2\mathrm{e}^{\lambda_{\mathrm{R}}t}\left(c_{\mathrm{R}}\cos(\lambda_{\mathrm{I}}t) - c_{\mathrm{I}}\sin(\lambda_{\mathrm{I}}t)\right) \\ &\overset{\text{[Apo67, S. 334]}}{=} 2\mathrm{e}^{\lambda_{\mathrm{R}}t}\operatorname{sgn}(c_{\mathrm{R}})\sqrt{c_{\mathrm{R}}^2 + c_{\mathrm{I}}^2}\cos\left(\lambda_{\mathrm{I}}t + \arctan\left(\frac{c_{\mathrm{I}}}{c_{\mathrm{R}}}\right)\right). \end{aligned} \tag{4.11}$$

Mit

$$\boldsymbol{\zeta}_{ij}(t) \coloneqq \begin{bmatrix} x_{\mathrm{r},ij}(t) & \dot{x}_{\mathrm{r},ij}(t) & \ldots & x_{\mathrm{r},ij}^{(\nu_{ij}-1)}(t) \end{bmatrix}^{\mathsf{T}} \tag{4.12}$$

lässt sich die gewöhnliche Differenzialgleichung (4.8) durch

$$\dot{\boldsymbol{\zeta}}_{ij}(t) = \boldsymbol{D}_{ij}\boldsymbol{\zeta}_{ij}(t) \tag{4.13}$$

darstellen, wobei

$$\boldsymbol{D}_{ij} := \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_{ij0} & -a_{ij1} & -a_{ij2} & \dots & -a_{ij(\nu_{ij}-1)} \end{bmatrix} \tag{4.14}$$

gilt. Werden die Vektoren $\boldsymbol{\zeta}_{ij}(t)$ zu

$$\boldsymbol{\zeta}_i(t) := \begin{bmatrix} \boldsymbol{\zeta}_{i1}^\mathsf{T}(t) & \dots & \boldsymbol{\zeta}_{iJ_i}^\mathsf{T}(t) \end{bmatrix}^\mathsf{T} \tag{4.15}$$

zusammengefasst, folgt

$$\boldsymbol{x}_\mathrm{r}(t) = \underbrace{\begin{bmatrix} \boldsymbol{s}_1^\mathsf{T} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{s}_2^\mathsf{T} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{s}_n^\mathsf{T} \end{bmatrix}}_{=:\boldsymbol{S}} \underbrace{\begin{bmatrix} \boldsymbol{\zeta}_1(t) \\ \boldsymbol{\zeta}_2(t) \\ \vdots \\ \boldsymbol{\zeta}_n(t) \end{bmatrix}}_{=:\boldsymbol{\zeta}(t)}, \tag{4.16}$$

wobei

$$\boldsymbol{s}_i^\mathsf{T} := \begin{bmatrix} \boldsymbol{s}_{i1}^\mathsf{T} & \dots & \boldsymbol{s}_{iJ_i}^\mathsf{T} \end{bmatrix} \tag{4.17}$$

und

$$\boldsymbol{s}_{ij} := \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{\nu_{ij}}, \forall i, j, \tag{4.18}$$

gilt. Zudem folgt

$$\dot{\boldsymbol{\zeta}}(t) = \boldsymbol{D}\boldsymbol{\zeta}(t) \tag{4.19}$$

mit

$$\boldsymbol{D} := \mathrm{diag}\left(\boldsymbol{D}_{11}, \dots, \boldsymbol{D}_{1J_1}, \dots, \boldsymbol{D}_{nJ_n}\right). \tag{4.20}$$

Für feste Hyperparameter λ_{ij} und ν_{ij} ist der Verlauf der Solltrajektorie $\boldsymbol{x}_\mathrm{r}(t)$ durch

$$\boldsymbol{c} := \begin{bmatrix} \boldsymbol{c}_{11}^\mathsf{T} & \dots & \boldsymbol{c}_{1J_1}^\mathsf{T} & \dots & \boldsymbol{c}_{nJ_n}^\mathsf{T} \end{bmatrix}^\mathsf{T} \tag{4.21}$$

mit

$$\boldsymbol{c}_{ij} := \begin{bmatrix} c_{ij0} & c_{ij1} & \dots & c_{ij(\nu_{ij}-1)} \end{bmatrix}^\mathsf{T} \tag{4.22}$$

parametriert.

Im Folgenden wird gezeigt, dass eine Bijektion zwischen $\boldsymbol{c}$ und $\boldsymbol{\zeta}(0)$ existiert, die Parametrierung der Solltrajektorie also eindeutig durch $\boldsymbol{\zeta}(0)$ codiert ist. Hierzu wird zunächst das folgende Hilfslemma benötigt.

Lemma 4.2

Die l-te zeitliche Ableitung von $x_{\mathrm{r},ij}(t)$ ist durch

$$x^{(l)}_{\mathrm{r},ij}(t) = \sum_{m=0}^{l} e^{\lambda_{ij} t} \binom{l}{m} \lambda_{ij}^{l-m} \sum_{k=m}^{\nu_{ij}-1} \frac{k!}{(k-m)!} c_{ijk} t^{k-m} \tag{4.23}$$

gegeben.

Beweis:
Siehe Anhang B.2. □

Proposition 4.2

Es existiert eine bijektive Abbildung zwischen $\boldsymbol{c}$ und $\boldsymbol{\zeta}(0)$ der Form

$$\boldsymbol{\zeta}(0) = \boldsymbol{\Gamma}\boldsymbol{c} \tag{4.24}$$

mit $\boldsymbol{\Gamma} \in \mathbb{R}^{n_\zeta \times n_\zeta}$.

Beweis:
Aus Lemma 4.2 folgt für $t = 0$

$$x^{(l)}_{\mathrm{r},ij}(0) = \sum_{m=0}^{l} \binom{l}{m} \lambda_{ij}^{l-m}\, m!\, c_{ijm} = \sum_{m=0}^{l} \frac{l!}{(l-m)!} \lambda_{ij}^{l-m} c_{ijm}. \tag{4.25}$$

Aus (4.12) ergibt sich der Initialzustand $\boldsymbol{\zeta}_{ij}(0)$ zu

$$\boldsymbol{\zeta}_{ij}(0) = \boldsymbol{\Gamma}_{ij}\boldsymbol{c}_{ij}, \tag{4.26}$$

wobei

$$(\boldsymbol{\Gamma}_{ij})_{lm} := \begin{cases} \frac{(l-1)!}{(l-m)!}\lambda_{ij}^{l-m}, & m \leq l \\ 0, & m > l \end{cases}, \quad l, m = 1, \ldots, \nu_{ij}, \tag{4.27}$$

gilt. Mithilfe des Laplaceschen Entwicklungssatzes [BHWM12, Satz 3.21] folgt, da es sich bei $\boldsymbol{\Gamma}_{ij}$ um eine untere Dreiecksmatrix handelt,

$$\det(\boldsymbol{\Gamma}_{ij}) = \prod_{l=1}^{\nu_{ij}} (\boldsymbol{\Gamma}_{ij})_{ll} = \prod_{l=0}^{\nu_{ij}-1} l! > 0. \tag{4.28}$$

Mit

$$\boldsymbol{\Gamma} := \operatorname{diag}(\boldsymbol{\Gamma}_{11}, \ldots, \boldsymbol{\Gamma}_{1J_1}, \ldots, \boldsymbol{\Gamma}_{nJ_n}) \tag{4.29}$$

ergibt sich nach [Pow11]

$$\det(\boldsymbol{\Gamma}) = \prod_{i=1}^{n} \prod_{j=1}^{J_i} \boldsymbol{\Gamma}_{ij} > 0. \tag{4.30}$$

Somit ist $\boldsymbol{\Gamma}$ regulär und stellt nach [Axl15, Theorem 10.24] eine Bijektion dar. □

Die nachfolgende Bemerkung begründet, weshalb $\boldsymbol{\zeta}(t)$ in (4.16) und $\boldsymbol{D}$ in (4.20) als reellwertig angenommen werden können.

Bemerkung 4.2

Zwar stellt Lemma 4.1 $\boldsymbol{x}_{\mathrm{r}}(t) \in \mathbb{R}^n$ sicher, dennoch können durch konjugiert-komplexe Polpaare λ_{ij} und $\lambda_{ij}^{\star} = \lambda_{ij^{\star}}$ komplexwertige Zustände $\boldsymbol{\zeta}_{ij}$ und $\boldsymbol{\zeta}_{ij^{\star}}$ sowie komplexwertige Einträge in den Elementen von $\boldsymbol{D}_{ij}$ bzw. $\boldsymbol{D}_{ij^{\star}}$ (vgl. (4.14)) resultieren. Durch eine eineindeutige Zustandstransformation lassen sich jedoch stets reellwertige Ersatzzustände mit einer reellen Dynamik erzeugen. Dieses Vorgehen ist in Anhang B.3 skizziert. Daher wird o. B. d. A. stets $\boldsymbol{\zeta}(t)$ in (4.16) und $\boldsymbol{D}$ in (4.20) als reellwertig angenommen.

Schließlich sei zu betonen, dass die soeben vorgestellte Referenztrajektoriendarstellung ADP-kompatibel ist. Dies ist in der nachfolgenden Bemerkung zusammengefasst.

Bemerkung 4.3

Die in diesem Abschnitt vorgestellte Referenztrajektoriendarstellung ist ADP-kompatibel im Sinne von Definition 4.1 mit

$$\boldsymbol{f}_{\boldsymbol{x}_{\mathrm{r}},\boldsymbol{\zeta}}(\boldsymbol{\zeta}(t)) \coloneqq \boldsymbol{S}\boldsymbol{\zeta}(t) \tag{4.31}$$

(vgl. (4.1)) und

$$\boldsymbol{f}_{\boldsymbol{\zeta}}(\boldsymbol{\zeta}(t)) \coloneqq \boldsymbol{D}\boldsymbol{\zeta}(t) \tag{4.32}$$

(vgl. (4.2)). Die Wahl der Hyperparameter λ_{ij} und ν_{ij} definiert die Klasse der Referenztrajektorienapproximation (vgl. Bemerkung 4.1). Zur Laufzeit kann der konkrete Solltrajektorienverlauf durch Vorgabe des Parameters $\mathbf{c}$ schließlich von außen beeinflusst werden.

4.2.2 Trajektorienfolgeregelung mit global diskontiertem Gütemaß

Bevor ein ADP-basierter Solltrajektorienfolgeregler entworfen wird, soll untersucht werden, unter welchen Bedingungen und für welche Gütemaße unter Verwendung der in Abschnitt 4.2.1 vorgestellten ADP-kompatiblen Referenztrajektoriendarstellung stabilisierende optimale Lösungen existieren.

In Anlehnung an die Eingangs-Zustands-Stabilität [Kha02, Definition 4.7] und analog zur zeitdiskreten Referenz-Zustands-Stabilität (vgl. Definition 3.5) wird hierfür zunächst der Begriff der zeitkontinuierlichen Referenz-Zustands-Stabilität definiert.

Definition 4.2 (Zeitkontinuierliche Referenz-Zustands-Stabilität)
Ein geregeltes System ist Referenz-Zustands-stabil, wenn für endliche Solltrajektorienparameter

$$\|\boldsymbol{c}\|_2 < \infty$$

beschränkte Systemzustände

$$\|\boldsymbol{x}(t)\|_2 < \infty,\ \forall t \geq 0,$$

resultieren.

Bei der Wahl eines geeigneten Gütemaßes, das einerseits die Abweichung des Systemzustands $\boldsymbol{x}(t)$ vom Sollzustand $\boldsymbol{x}_\mathrm{r}(t)$ und andererseits die durch die Stellgröße $\boldsymbol{u}(t)$ aufgebrachte Stellenergie bestraft, ist ein schlecht gestelltes, d. h. mit unendlichen Gesamtkosten verbundenes, Problem zu vermeiden (vgl. [XZLJ16]). Unendlich hohe Gesamtkosten entstünden beispielsweise dann, wenn das Halten des Systemzustands in einem Sollzustand mit Stellenergie verbunden wäre, und Stellenergie sowie die Abweichung vom Sollzustand im Gütemaß mit unendlichem Optimierungshorizont bestraft würden. Daher ist die Verwendung eines globalen Diskontierungsfaktors γ, wobei für $\gamma > 0$ in der Zukunft liegende Kosten gedämpft werden, ein in der RL- und ADP-Literatur häufig verwendeter Ansatz (vgl. Tabelle 2.2). Für

$$\dot{\tilde{\boldsymbol{x}}}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\boldsymbol{x}(t)\\ \boldsymbol{\zeta}(t)\end{bmatrix} = \begin{bmatrix}\boldsymbol{A} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{D}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}(t)\\ \boldsymbol{\zeta}(t)\end{bmatrix} + \begin{bmatrix}\boldsymbol{B}\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{u}(t) =: \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}(t) + \tilde{\boldsymbol{B}}\boldsymbol{u}(t) \tag{4.33}$$

werde zunächst das folgende Problem betrachtet[78].

[78] Grundsätzlich lässt sich die in Abschnitt 4.2 vorgestellte ADP-kompatible Solltrajektorienrepräsentation auch auf nichtlineare Systeme anwenden. Für eingangsaffine Systeme ergibt sich beispielsweise die erweiterte Systemdynamik $\dot{\tilde{\boldsymbol{x}}}(t) := \begin{bmatrix}\boldsymbol{f}(\boldsymbol{x}(t))\\ \boldsymbol{D}\boldsymbol{\zeta}(t)\end{bmatrix} + \begin{bmatrix}\boldsymbol{g}(\boldsymbol{x}(t))\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{u}(t) =: \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}(t)) + \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}}(t))\boldsymbol{u}(t)$ sowie die Value Function $V^{\boldsymbol{\mu}}(\boldsymbol{x},\boldsymbol{\zeta}) = \int_t^\infty \mathrm{e}^{-\gamma(\tau-t)}\left(q(\boldsymbol{x},\boldsymbol{\zeta}) + \boldsymbol{\mu}^\intercal\boldsymbol{R}\boldsymbol{\mu}\right)\,\mathrm{d}\tau$. Jedoch ist die Form der Value Function $V^{\boldsymbol{\mu}}$ in diesem Fall im Allgemeinen unbekannt und die Wahl geeigneter Basisfunktionen zur Funktionsapproximation (vgl. Abschnitt 2.1.3) nach wie vor ungelöst [WHL17]. Zwar erlaubt beispielsweise der Satz von Stone-Weierstraß Aussagen zur Approximierbarkeit stetiger Funktionen auf einer kompakten Menge durch genügend viele Basisfunktionen (vgl. [Che78, S. 226], [Wei85]), jedoch kann die Anzahl benötigter Basisfunktionen gegebenenfalls hoch sein und die lediglich lokale Gültigkeit muss berücksichtigt werden.

Problem 4.1 (Globale Diskontierung)

$$\min_{\boldsymbol{\mu}(\boldsymbol{x}(\tau),\boldsymbol{\zeta}(\tau))} \int_t^\infty e^{-\gamma(\tau-t)} \Big(\tilde{\boldsymbol{x}}^\mathsf{T}(\tau)\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{x}}(\tau) + \boldsymbol{\mu}^\mathsf{T}\big(\boldsymbol{x}(\tau),\boldsymbol{\zeta}(\tau)\big)\boldsymbol{R}\boldsymbol{\mu}\big(\boldsymbol{x}(\tau),\boldsymbol{\zeta}(\tau)\big)\Big)\,\mathrm{d}\tau \tag{4.34}$$

$$\text{u. d. N.}\quad \dot{\tilde{\boldsymbol{x}}}(\tau) = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}(\tau) + \tilde{\boldsymbol{B}}\boldsymbol{\mu}\big(\boldsymbol{x}(\tau),\boldsymbol{\zeta}(\tau)\big) \tag{4.35}$$

mit

$$\tilde{\boldsymbol{Q}} := \begin{bmatrix}\boldsymbol{I} & -\boldsymbol{S}\end{bmatrix}^\mathsf{T} \boldsymbol{Q} \begin{bmatrix}\boldsymbol{I} & -\boldsymbol{S}\end{bmatrix}. \tag{4.36}$$

Mit (4.16) ergibt sich

$$\begin{aligned}\tilde{\boldsymbol{x}}^\mathsf{T}(\tau)\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{x}}(\tau) &= \begin{bmatrix}\boldsymbol{x}(\tau) & \boldsymbol{\zeta}(\tau)\end{bmatrix} \begin{bmatrix}\boldsymbol{I} \\ -\boldsymbol{S}\end{bmatrix} \boldsymbol{Q} \begin{bmatrix}\boldsymbol{I} & -\boldsymbol{S}\end{bmatrix} \begin{bmatrix}\boldsymbol{x}(\tau) \\ \boldsymbol{\zeta}(\tau)\end{bmatrix} \\ &= \left(\boldsymbol{x}(\tau) - \boldsymbol{x}_\mathrm{r}(\tau)\right)^\mathsf{T} \boldsymbol{Q} \left(\boldsymbol{x}(\tau) - \boldsymbol{x}_\mathrm{r}(\tau)\right),\end{aligned} \tag{4.37}$$

d. h. in (4.34) wird die Abweichung des Zustands $\boldsymbol{x}(\tau)$ von $\boldsymbol{x}_\mathrm{r}(\tau)$ bestraft. Problem 4.1 kann in das folgende äquivalente Ersatzproblem überführt werden. Die Äquivalenz ist in Anhang B.4 gezeigt.

Problem 4.2 (Äquivalente Formulierung zur globalen Diskontierung)

$$\min_{\boldsymbol{\mu}(\boldsymbol{x}(\tau),\boldsymbol{\zeta}(\tau))} \int_t^\infty \Big(\tilde{\boldsymbol{x}}^\mathsf{T}(\tau)\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{x}}(\tau) + \boldsymbol{\mu}^\mathsf{T}\big(\boldsymbol{x}(\tau),\boldsymbol{\zeta}(\tau)\big)\boldsymbol{R}\boldsymbol{\mu}\big(\boldsymbol{x}(\tau),\boldsymbol{\zeta}(\tau)\big)\Big)\,\mathrm{d}\tau \tag{4.38}$$

$$\text{u. d. N.}\quad \dot{\tilde{\boldsymbol{x}}}(\tau) = \Big(\tilde{\boldsymbol{A}} - \frac{\gamma}{2}\boldsymbol{I}\Big)\tilde{\boldsymbol{x}}(\tau) + \tilde{\boldsymbol{B}}\boldsymbol{\mu}\big(\boldsymbol{x}(\tau),\boldsymbol{\zeta}(\tau)\big) \tag{4.39}$$

mit $\tilde{\boldsymbol{Q}}$ *wie in* (4.36).

Somit führt die Verwendung eines globalen Diskontierungsfaktors γ zum gleichen Effekt, als würden bei einem nicht-diskontierten Optimierungsproblem alle Eigenwerte der erweiterten Systemmatrix $\tilde{\boldsymbol{A}}$ um $\frac{\gamma}{2}$ nach links verschoben. Diese Verschiebung der Eigenwerte findet jedoch nur virtuell statt, d. h. aus Sicht des zu minimierenden global diskontierten Gütefunktionals wirken die Systemeigenwerte verschoben, das reale System bleibt jedoch nach wie vor durch (4.33) gegeben. Dies kann potenziell dazu führen, dass eine für das Ersatzproblem 4.2 stabilisierende und optimale Lösung

$$\boldsymbol{\mu}^*\big(\boldsymbol{x}(t),\boldsymbol{\zeta}(t)\big) = -\tilde{\boldsymbol{K}}^*\tilde{\boldsymbol{x}}(t) = -\begin{bmatrix}\boldsymbol{K}_\mathrm{x}^* & \boldsymbol{K}_\zeta^*\end{bmatrix}\begin{bmatrix}\boldsymbol{x}(t) \\ \boldsymbol{\zeta}(t)\end{bmatrix} \tag{4.40}$$

gefunden wird, die zwar aufgrund der Äquivalenz zu Problem 4.1 auch (4.34) minimiert, jedoch mit einem instabilen Systemverhalten des Systems (4.35) und damit auch des Systems

$$\dot{\boldsymbol{x}}(t) = \left(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_{\mathrm{x}}^{*}\right)\boldsymbol{x}(t) \tag{4.41}$$

einhergeht. Dies wird anhand des nachfolgenden Beispiels illustriert.

Beispiel 4.1 *Betrachtet werde das stabilisierbare System zweiter Ordnung mit der Systemdynamik*

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 0 & 1 \\ -1{,}01 & 0{,}2 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t). \tag{4.42}$$

Für eine ADP-kompatible Solltrajektoriendarstellung nach (4.7) *werde ein einzelner Eigenwert bei* $\lambda_{11} = 0$ *gewählt, wodurch die Vorgabe eines konstanten Sollzustands für den Zustand* $x_1(t)$ *ermöglicht wird. Dies führt nach* (4.39) *auf die effektive erweiterte Systemdynamik*

$$\dot{\tilde{\boldsymbol{x}}}(t) = \underbrace{\begin{bmatrix} -\frac{\gamma}{2} & 1 & 0 \\ -1{,}01 & 0{,}2 - \frac{\gamma}{2} & 0 \\ 0 & 0 & -\frac{\gamma}{2} \end{bmatrix}}_{=\left(\tilde{\boldsymbol{A}} - \frac{\gamma}{2}\boldsymbol{I}\right)} \tilde{\boldsymbol{x}}(t) + \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}}_{\tilde{\boldsymbol{B}}} u(t). \tag{4.43}$$

Mit

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \boldsymbol{S} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{R} = 100 \ \textit{ und } \ \gamma = 1 \tag{4.44}$$

existiert eine optimale Lösung für Problem 4.2 und das optimale Regelgesetz ergibt sich zu

$$\boldsymbol{\mu}(\tilde{\boldsymbol{x}}(t)) = -\underbrace{\begin{bmatrix} 0{,}0067 & 0{,}0016 & -0{,}0044 \end{bmatrix}}_{=\tilde{\boldsymbol{K}}^{*}} \tilde{\boldsymbol{x}}(t). \tag{4.45}$$

Die Systempole des mittels (4.45) *geschlossenen Regelkreises, d. h. die Eigenwerte von* $\tilde{\boldsymbol{A}} - \tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}^{*}$ *ergeben sich zu* $0{,}0966 \pm \mathrm{j}0{,}9988; 0$. *Dies führt selbst bei der Vorgabe* $x_{\mathrm{r},1}(t) = 0$ *zu einem instabilen System, wie in Abbildung 4.1 für* $\tilde{\boldsymbol{x}}(0) = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^{\mathsf{T}}$ *veranschaulicht ist. Die mögliche Instabilität ist somit auf die globale Diskontierung und nicht auf die Solltrajektoriendarstellung zurückzuführen. Insbesondere ist das System für dieses Beispiel auch nicht Referenz-Zustands-stabil nach Definition 4.2.*

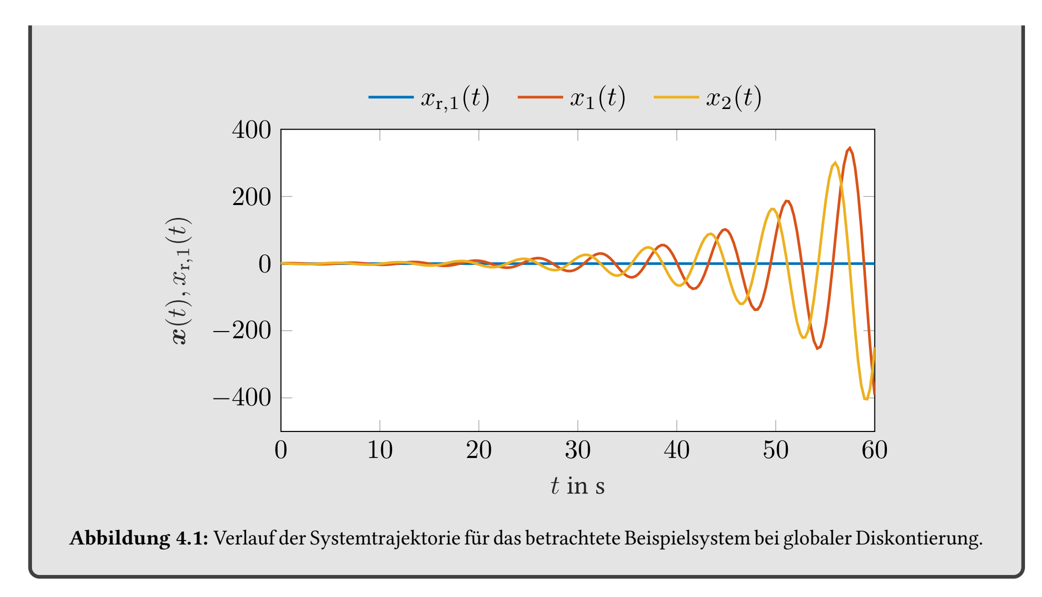


Abbildung 4.1: Verlauf der Systemtrajektorie für das betrachtete Beispielsystem bei globaler Diskontierung.

Da durch die Verwendung einer globalen Diskontierung $\gamma > 0$ und die damit verbundene Dämpfung der Kosten in (4.38), wie in Beispiel 4.2 ersichtlich, ein instabiles Regelgesetz resultieren kann, wird im nächsten Abschnitt die Verwendung eines Diskontierungsfaktors, der lediglich die Dynamik $\boldsymbol{D}$ dämpft, vorgestellt.

4.2.3 Trajektorienfolgeregelung mit gedämpfter Referenzdynamik

Um trotz einer gegebenenfalls nicht asymptotisch stabilen Solltrajektoriendynamik $\boldsymbol{D}$ nach (4.19) und ohne Verwendung einer globalen Diskontierung wie in Problem 4.1 ein Optimierungsproblem mit endlichen Kosten zu definieren, wird die Verwendung einer teilweisen Dämpfung der erweiterten Systemdynamik gemäß der nachfolgenden Problemstellung betrachtet.

Problem 4.3 (Teilweise Dämpfung)

$$\min_{\boldsymbol{\mu}(\boldsymbol{x}(\tau),\boldsymbol{\zeta}(\tau))} \int_t^{\infty} \Big(\tilde{\boldsymbol{x}}^{\mathsf{T}}(\tau)\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{x}}(\tau) + \boldsymbol{\mu}^{\mathsf{T}}\big(\boldsymbol{x}(\tau),\boldsymbol{\zeta}(\tau)\big)\boldsymbol{R}\boldsymbol{\mu}\big(\boldsymbol{x}(\tau),\boldsymbol{\zeta}(\tau)\big)\Big)\,\mathrm{d}\tau \tag{4.46}$$

$$\text{u. d. N.}\;\; \dot{\tilde{\boldsymbol{x}}}(\tau) = \tilde{\boldsymbol{A}}'\tilde{\boldsymbol{x}}(\tau) + \tilde{\boldsymbol{B}}\boldsymbol{\mu}\big(\boldsymbol{x}(\tau),\boldsymbol{\zeta}(\tau)\big) \tag{4.47}$$

mit

$$\tilde{\boldsymbol{A}}' := \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D} - \frac{\gamma}{2}\boldsymbol{I} \end{bmatrix} =: \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}' \end{bmatrix} \tag{4.48}$$

und $\tilde{\boldsymbol{Q}}$ *wie in* (4.36).

Bemerkung 4.4
Für den mit der gedämpften Dynamik $\boldsymbol{D}' = \boldsymbol{D} - \frac{\gamma}{2}\boldsymbol{I}$ *propagierten Trajektorienparameter* $\boldsymbol{\zeta}(t)$ *folgt mit*

$$\boldsymbol{f}_{\boldsymbol{\zeta}}(\boldsymbol{\zeta}(t)) \coloneqq \boldsymbol{D}'\boldsymbol{\zeta}(t) \tag{4.49}$$

analog zu Bemerkung 4.3 ADP-Kompatibilität.

Zudem werden die folgenden Standardannahmen für LQ-Optimierungsprobleme (vgl. [AM89, Abschnitt 3.2], [LVS12, Abschnitt 3.4], [Kuč73]) vorausgesetzt.

Annahme 4.2

1. $(\boldsymbol{A}, \boldsymbol{B})$ *sei stabilisierbar.*
2. *Seien* $\boldsymbol{Q} \succeq \boldsymbol{0}$ *und* $\boldsymbol{R} \succ \boldsymbol{0}$.
3. $(\boldsymbol{A}, \sqrt{\boldsymbol{Q}})$ *sei detektierbar, wobei* $\sqrt{\boldsymbol{Q}}^\mathsf{T}\sqrt{\boldsymbol{Q}} = \boldsymbol{Q}$ *gilt.*

Im Folgenden wird die Stabilität des geschlossenen Regelkreises für die nach Problem 4.3 optimale Lösung untersucht. Bevor durch Satz 4.1 die Hauptaussage des vorliegenden Abschnitts folgt, werden zunächst einige hierfür benötigte Zusammenhänge gegeben.

Lemma 4.3
Seien

$$\boldsymbol{M}_1 \in \mathbb{R}^{n_1 \times m_1}, \quad n_1 \leq m_1, \tag{4.50}$$
$$\boldsymbol{M}_2 \in \mathbb{R}^{n_2 \times m_2}, \quad n_2 \leq m_2 \tag{4.51}$$

Matrizen mit Maximalrang, d. h. es gelte

$$\text{Rang}\,(\boldsymbol{M}_1) = \text{Dim}(\text{Bild}(\boldsymbol{M}_1)) = n_1, \tag{4.52}$$
$$\text{Rang}\,(\boldsymbol{M}_2) = \text{Dim}(\text{Bild}(\boldsymbol{M}_2)) = n_2. \tag{4.53}$$

Dann weist

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_1 & \boldsymbol{M}_3 \\ \boldsymbol{0} & \boldsymbol{M}_2 \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times(m_1+m_2)} \tag{4.54}$$

für beliebige $\boldsymbol{M}_3 \in \mathbb{R}^{n_1 \times m_2}$ *den Maximalrang*

$$\text{Rang}\,(\boldsymbol{M}) = n_1 + n_2 \tag{4.55}$$

auf.

Beweis:
Der Beweis ist in Anhang B.5 gegeben. □

Im weiteren Verlauf sei

$$\gamma_{\min} := 2 \max\left(0, \lambda^{+}_{\mathrm{Re}}(\boldsymbol{D})\right), \tag{4.56}$$

wobei $\lambda^{+}_{\mathrm{Re}}(\boldsymbol{D})$ dem Eigenwert von $\boldsymbol{D}$ mit größtem Realteil entspricht[79]. Damit lässt sich das folgende Lemma formulieren.

Lemma 4.4
Unter Annahme 4.2 ist das durch $\left(\tilde{\boldsymbol{A}}', \tilde{\boldsymbol{B}}\right)$ *beschriebene System mit* $\tilde{\boldsymbol{A}}'$ *wie in* (4.48) *und* $\tilde{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{0} \end{bmatrix}$ *stabilisierbar, falls* $\gamma > \gamma_{\min}$ *gilt.*

Beweis:
Siehe Anhang B.6. □

Lemma 4.5
Seien $\boldsymbol{M}_1, \boldsymbol{M}_2$ *wie in* (4.50)–(4.53). *Dann weist*

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_1 & \boldsymbol{0} \\ \boldsymbol{M}_3 & \boldsymbol{M}_2 \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times(m_1+m_2)} \tag{4.57}$$

für beliebige $\boldsymbol{M}_3 \in \mathbb{R}^{n_2 \times m_1}$ *den Maximalrang*

$$\mathrm{Rang}\left(\boldsymbol{M}\right) = n_1 + n_2 \tag{4.58}$$

auf.

Beweis:
Der Beweis erfolgt analog zum Beweis von Lemma 4.3. □

Lemma 4.6
Unter Annahme 4.2 ist $\left(\tilde{\boldsymbol{A}}', \sqrt{\tilde{\boldsymbol{Q}}}\right)$ *detektierbar, falls* $\gamma > \gamma_{\min}$ *gilt. Hierbei sei* $\tilde{\boldsymbol{A}}'$ *wie in* (4.48) *und* $\sqrt{\tilde{\boldsymbol{Q}}}^{\mathsf{T}} \sqrt{\tilde{\boldsymbol{Q}}} = \tilde{\boldsymbol{Q}}$ *mit* $\tilde{\boldsymbol{Q}}$ *aus* (4.36).

[79] Nach Bemerkung 4.2 ist $\boldsymbol{D}$ reellwertig.

Beweis:
Siehe Anhang B.7. □

Satz 4.1
Sei Annahme 4.2 erfüllt, zudem gelte $\gamma > \gamma_{min} = 2\max\big(0, \lambda_{Re}^{+}(\boldsymbol{D})\big)$. *Dann existiert eine eindeutige, symmetrische, positiv semidefinite Lösung* $\tilde{\boldsymbol{P}}^*$ *der algebraischen Riccati-Gleichung*

$$\tilde{\boldsymbol{Q}} + \tilde{\boldsymbol{P}}^*\tilde{\boldsymbol{A}}' + \tilde{\boldsymbol{A}}'^{\mathsf{T}}\tilde{\boldsymbol{P}}^* - \tilde{\boldsymbol{P}}^*\tilde{\boldsymbol{B}}\boldsymbol{R}^{-1}\tilde{\boldsymbol{B}}^{\mathsf{T}}\tilde{\boldsymbol{P}}^* = \boldsymbol{0}. \tag{4.59}$$

Zudem ist

$$\boldsymbol{\mu}^*\big(\boldsymbol{x}(t), \boldsymbol{\zeta}(t)\big) = -\tilde{\boldsymbol{K}}^*\tilde{\boldsymbol{x}}(t) = -\begin{bmatrix}\boldsymbol{K}_{\mathrm{x}}^* & \boldsymbol{K}_{\zeta}^*\end{bmatrix}\begin{bmatrix}\boldsymbol{x}(t)\\ \boldsymbol{\zeta}(t)\end{bmatrix} \tag{4.60}$$

mit

$$\tilde{\boldsymbol{K}}^* = -\boldsymbol{R}^{-1}\tilde{\boldsymbol{B}}^{\mathsf{T}}\tilde{\boldsymbol{P}}^* \tag{4.61}$$

optimal bezüglich Problem 4.3 und

$$\dot{\tilde{\boldsymbol{x}}}(t) = \left(\tilde{\boldsymbol{A}}' - \tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}^*\right)\tilde{\boldsymbol{x}}(t) \tag{4.62}$$

und

$$\dot{\boldsymbol{x}}(t) = \left(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_{\mathrm{x}}^*\right)\boldsymbol{x}(t) \tag{4.63}$$

sind global asymptotisch stabil.

Beweis:
Nach Annahme 4.2 gilt $\boldsymbol{R} \succ \boldsymbol{0}$ und $\boldsymbol{Q} \succeq \boldsymbol{0}$. Somit folgt aufgrund von

$$\tilde{\boldsymbol{x}}^{\mathsf{T}}\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{x}} = \big(\begin{bmatrix}\boldsymbol{I} & -\boldsymbol{S}\end{bmatrix}\tilde{\boldsymbol{x}}\big)^{\mathsf{T}}\boldsymbol{Q}\big(\begin{bmatrix}\boldsymbol{I} & -\boldsymbol{S}\end{bmatrix}\tilde{\boldsymbol{x}}\big) \geq 0, \quad \forall\tilde{\boldsymbol{x}}, \tag{4.64}$$

direkt $\tilde{\boldsymbol{Q}} \succeq \boldsymbol{0}$. Zudem ist $\left(\tilde{\boldsymbol{A}}', \tilde{\boldsymbol{B}}\right)$ stabilisierbar (Lemma 4.4) und $\left(\tilde{\boldsymbol{A}}', \sqrt{\tilde{\boldsymbol{Q}}}\right)$ detektierbar (Lemma 4.6). Existenz und Eindeutigkeit von $\tilde{\boldsymbol{P}}^*$ folgen daher direkt aus [AM89, Abschnitt 3.2] und [Kuč73, Theorem 5], ebenso die global asymptotische Stabilität von (4.62).

Schließlich wird global asymptotische Stabilität von (4.63) gezeigt. Wegen

$$\tilde{\boldsymbol{A}}' - \tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}^* = \begin{bmatrix}\boldsymbol{A} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{D}'\end{bmatrix} - \begin{bmatrix}\boldsymbol{B}\\ \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{K}_{\mathrm{x}}^* & \boldsymbol{K}_{\zeta}^*\end{bmatrix} = \begin{bmatrix}\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_{\mathrm{x}}^* & -\boldsymbol{B}\boldsymbol{K}_{\zeta}^*\\ \boldsymbol{0} & \boldsymbol{D}'\end{bmatrix} \tag{4.65}$$

folgt mit [Pow11]

$$\det\left(\lambda\boldsymbol{I}-\left(\tilde{\boldsymbol{A}}'-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}^*\right)\right)=\det\bigl(\lambda\boldsymbol{I}-(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_{\mathrm{x}}^*)\bigr)\det\left(\lambda\boldsymbol{I}-\boldsymbol{D}'\right). \tag{4.66}$$

Eigenwerte von $\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_{\mathrm{x}}^*$ sind somit zugleich auch Eigenwerte von $\left(\tilde{\boldsymbol{A}}'-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}^*\right)$, weshalb global asymptotische Stabilität von (4.63) resultiert. □

Korollar 4.1
Das System

$$\dot{\boldsymbol{x}}(t)=(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_{\mathrm{x}}^*)\,\boldsymbol{x}(t)-\boldsymbol{B}\boldsymbol{K}_{\zeta}^*\boldsymbol{\Gamma}\boldsymbol{c} \tag{4.67}$$

ist Referenz-Zustands-stabil nach Definition 4.2.

Beweis:
Da (4.63) nach Satz 4.1 global asymptotisch stabil ist, ist (4.67), wenn $(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}_{\mathrm{x}}^*)$ als Systemmatrix und $-\boldsymbol{B}\boldsymbol{K}_{\zeta}^*\boldsymbol{\Gamma}$ als Eingangsmatrix interpretiert wird, Eingangs-Ausgangs-stabil [Lun20b, S. 59]. Für endliche Solltrajektorienparameter $\boldsymbol{c}$ folgen somit stets beschränkte Systemzustände und (4.67) ist Referenz-Zustands-stabil. □

Dies zeigt insbesondere, dass, obwohl durch eine äußere Vorgabe von $\boldsymbol{c}$ eine Neuinitialisierung von $\boldsymbol{\zeta}$ erfolgt (vgl. Proposition 4.2), die Stabilität des Systemzustands $\boldsymbol{x}(t)$ durch die beliebige Vorgabe endlicher Solltrajektorienparameter $\boldsymbol{c}$ nicht gefährdet wird. Der in (4.67) verwendete Regler

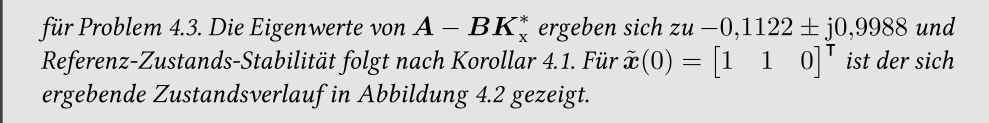
$$\boldsymbol{\mu}\bigl(\boldsymbol{x}(t),\boldsymbol{c}\bigr)=-\boldsymbol{K}_{\mathrm{x}}^*\boldsymbol{x}(t)-\boldsymbol{K}_{\zeta}^*\boldsymbol{\Gamma}\boldsymbol{c} \tag{4.68}$$

liefert dabei die nach Problem 4.3 optimale Stellgröße, wobei durch Wahl von $\boldsymbol{\Gamma}\boldsymbol{c}$ der Initialwert $\boldsymbol{\zeta}(t)$ in (4.46) eingestellt wird. Mithilfe von $\boldsymbol{c}$ kann somit jederzeit der lokale Verlauf der Solltrajektorie beeinflusst werden.

Das nachfolgende Beispiel greift das in Beispiel 4.2 bereits betrachtete System erneut auf und veranschaulicht, dass die in Problem 4.3 vorgestellte teilweise Dämpfung zu einem geregelten System führt, welches Referenz-Zustands-stabil ist.

Beispiel 4.2 *Betrachtet werde das durch* (4.42) *gegebene System, zudem sei* $\boldsymbol{D}=0$ *und somit* $\boldsymbol{D}'=-\frac{\gamma}{2}\boldsymbol{I}$. *Mit* $\boldsymbol{Q}$, $\boldsymbol{R}$, $\boldsymbol{S}$ *und* γ *wie in* (4.44) *folgt das optimale Regelgesetz*

$$\boldsymbol{\mu}(\tilde{\boldsymbol{x}}(t))=-\underbrace{\begin{bmatrix}0{,}4245 & -0{,}0843 & -0{,}0022\end{bmatrix}}_{=\tilde{\boldsymbol{K}}^*}\tilde{\boldsymbol{x}}(t) \tag{4.69}$$



für Problem 4.3. Die Eigenwerte von $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}_{\mathrm{x}}^{*}$ ergeben sich zu $-0{,}1122 \pm \mathrm{j}0{,}9988$ und Referenz-Zustands-Stabilität folgt nach Korollar 4.1. Für $\tilde{\boldsymbol{x}}(0) = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^{\mathsf{T}}$ ist der sich ergebende Zustandsverlauf in Abbildung 4.2 gezeigt.

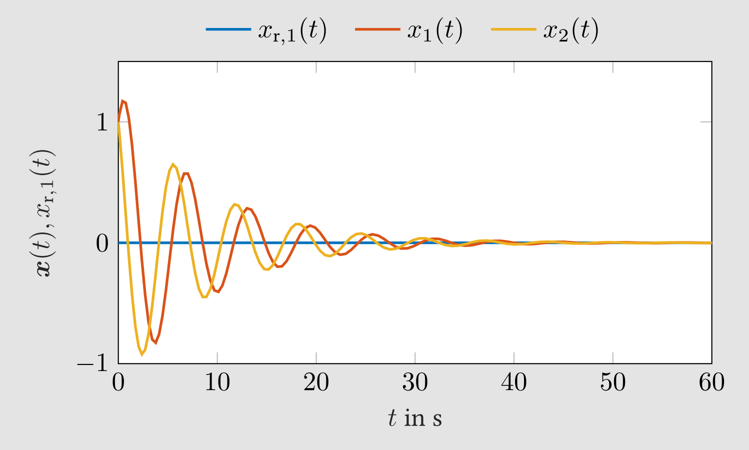


Abbildung 4.2: Verlauf der Systemtrajektorie für das betrachtete Beispielsystem bei teilweiser Dämpfung.

Für den Rest dieses Kapitels wird daher die in Problem 4.3 vorgestellte teilweise Dämpfung der erweiterten Systemdynamik verwendet.

Die nachfolgende Bemerkung zeigt jedoch, dass auch die Nutzung einer global diskontierten Gütefunktion prinzipiell möglich wäre, solange die Diskontierung nicht zu stark ist, d. h. γ nicht zu groß gewählt wird[80].

Bemerkung 4.5

Eine Möglichkeit, unter Annahme 4.2 stabilisierende Lösungen für LQ-Solltrajektorienfolgeregelungsprobleme zu gewährleisten, ist nach Satz 4.1 durch die Verwendung einer teilweise gedämpften erweiterten Systemdynamik nach Problem 4.3 gegeben.
Eine Alternative dazu ist die Verwendung einer global diskontierten Gütefunktion nach Problem 4.1 unter Beachtung zusätzlicher Nebenbedingungen bei der Wahl des Diskontierungsfaktors γ. Beispielsweise ist in [XZLJ16, Theorem 4] durch $\gamma \leq 2 \left\| \left(\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^{\mathsf{T}} \right)^{\frac{1}{2}} \right\|_2$ eine obere Schranke für global diskontierte LQ-Regulierungsprobleme gegeben, sodass eine stabilisierende Lösung existiert. Analog liefert [YWM+19, Theorem 1] obere Schranken für Zwei-Spieler-LQ-Differenzialspiele. Zur Berechnung dieser Schranken müssten jedoch sowohl die Parameter der jeweiligen Gütefunktionen als auch die jeweiligen Eingangsmatrizen $\boldsymbol{B}$ bekannt sein. Zu beachten ist weiterhin, dass $\boldsymbol{x}_{\mathrm{r}}(t)$ nicht zu unendlichen

80 Analog hierzu stellt Satz 3.3 im zeitdiskreten Fall Stabilität sicher, sofern dort die Diskontierung nicht zu stark gewählt wird, d. h. γ nicht zu klein gewählt wird.

Gesamtkosten führen darf. In [XZLJ16] wird daher beispielsweise zusätzlich zur oberen Schranke von γ gefordert, dass $\boldsymbol{x}_{\mathrm{r}}(t)$ beschränkt ist und $\gamma > 0$ gilt[81].
Die Verwendung einer globalen Diskontierung ist somit durchaus möglich, sofern die Diskontierung mit Bedacht gewählt wird. Dies ist insbesondere für Algorithmen relevant, deren Konvergenzaussagen von der Verwendung einer globalen Diskontierung profitieren[82].

Nachdem in diesem Abschnitt eine ADP-kompatible lokale Referenztrajektorienrepräsentation vorgestellt wurde, wird diese Solltrajektoriendarstellung im nachfolgenden Abschnitt für den Entwurf eines ADP-basierten Solltrajektorienreglers verwendet.

4.2.4 ADP-Umsetzung

In diesem Abschnitt werden die Struktur des ADP-basierten Solltrajektorienfolgereglers, der die in Abschnitt 4.2.1 präsentierte Trajektorienrepräsentation verwendet, und der zugrunde liegende ADP-Algorithmus, der genutzt wird, vorgestellt. Anschließend werden in Abschnitt 4.2.5 Simulationsergebnisse präsentiert.

Um Problem 4.3 basierend auf Messdaten und ohne Kenntnis der Werte von $\boldsymbol{A}$ und $\boldsymbol{B}$ lösen zu können, wird eine zeitkontinuierliche Value Iteration (vgl. Abschnitt 2.1.4.2), konkret der Algorithmus von Bian und Jiang [BJ16a], verwendet. Dieser Ansatz benötigt keine explizite Kenntnis eines zulässigen initialen Regelgesetzes und gehört zudem zur Klasse der Off-Policy-Algorithmen inklusive der in Abschnitt 2.1.4.4 genannten Vorteile. Die zeitkontinuierliche Value Iteration [BJ16a] ist der Klasse der IRL-Algorithmen zuzuordnen und basiert auf einer Grenzwertbetrachtung der zeitdiskreten VI (vgl. Abschnitt 2.1.4.2). Nachfolgend stellt t die physikalische Zeit (und letztlich auch die Zeit der Datenaufzeichnung) dar, wohingegen s die Zeitvariable des Adaptionsvorgangs beschreibt (dies entspricht dem Iterationsindex l der zeitdiskreten VI). Das Ziel der Methode ist somit die Konvergenz

$$\lim_{s\to\infty} V^{\boldsymbol{\mu}_s}\big(\tilde{\boldsymbol{x}}(t)\big) = V^*\big(\tilde{\boldsymbol{x}}(t)\big). \tag{4.70}$$

Aus (2.20b) folgt

$$V^{\boldsymbol{\mu}_{s+T_{\mathrm{IRL}}}}\big(\tilde{\boldsymbol{x}}(t)\big) = \min_{\boldsymbol{\mu}}\left(\int_t^{t+T_{\mathrm{IRL}}} r(\tilde{\boldsymbol{x}},\boldsymbol{\mu})\,\mathrm{d}\tau + V^{\boldsymbol{\mu}_s}\big(\tilde{\boldsymbol{x}}(t+T_{\mathrm{IRL}})\big)\right) \tag{4.71}$$

81 Für eine instabile Exosystemdynamik würde sich, aufgrund der Nicht-Steuerbarkeit von $\boldsymbol{\zeta}(t)$, auch bei Verwendung einer globalen Diskontierung zusätzlich zu einer oberen Schranke für γ (vgl. [XZLJ16, Theorem 4]) die untere Schranke $\gamma_{\min}$ (4.56) ergeben.

82 Siehe beispielsweise [LP03, Theorem 7.1] für Konvergenzaussagen des zeitdiskreten LSPI-Ansatzes. In [HWL21, Theorem 2] wird zudem eine untere Schranke für die Wahl eines Diskontierungsfaktors eines global diskontierten zeitkontinuierlichen Optimierungsproblems gegeben, um Stabilität zu gewährleisten.

(vgl. (2.24)) und somit

$$\begin{aligned}\frac{\partial V^{\boldsymbol{\mu}_s}\big(\tilde{\boldsymbol{x}}(t)\big)}{\partial s} &= \min_{\boldsymbol{\mu}}\left(r\big(\tilde{\boldsymbol{x}}(t),\boldsymbol{\mu}\big) + \Big(\nabla_{\tilde{\boldsymbol{x}}} V^{\boldsymbol{\mu}_s}\big(\tilde{\boldsymbol{x}}(t)\big)\Big)^{\mathsf{T}}\frac{\mathrm{d}\tilde{\boldsymbol{x}}(t)}{\mathrm{d}t}\right)\\ &\overset{(2.14)}{=} \min_{\boldsymbol{\mu}} H\big(\tilde{\boldsymbol{x}}, \nabla_{\tilde{\boldsymbol{x}}} V^{\boldsymbol{\mu}_s}(\tilde{\boldsymbol{x}}), \boldsymbol{\mu}\big)\end{aligned} \tag{4.72}$$

(vgl. [BJ16a], [Büh20, Abschnitt 3.1.2]). Die Value Function $V^{\boldsymbol{\mu}_s}(\tilde{\boldsymbol{x}})$ sowie $\big(\nabla_{\tilde{\boldsymbol{x}}} V^{\boldsymbol{\mu}_s}(\tilde{\boldsymbol{x}})\big)^{\mathsf{T}}\frac{\mathrm{d}\tilde{\boldsymbol{x}}(t)}{\mathrm{d}t}$ werden entsprechend [BJ16a] durch lineare Funktionsapproximatoren (vgl. Abschnitt 2.1.3)

$$V^{\boldsymbol{\mu}_s}(\tilde{\boldsymbol{x}}) = \boldsymbol{\phi}^{\mathsf{T}}(\tilde{\boldsymbol{x}})\hat{\boldsymbol{w}}_{\phi}(s) \tag{4.73}$$

und

$$\big(\nabla_{\tilde{\boldsymbol{x}}} V^{\boldsymbol{\mu}_s}(\tilde{\boldsymbol{x}})\big)^{\mathsf{T}}\frac{\mathrm{d}\tilde{\boldsymbol{x}}(t)}{\mathrm{d}t} = \boldsymbol{\psi}^{\mathsf{T}}(\tilde{\boldsymbol{x}},\boldsymbol{u})\hat{\boldsymbol{w}}_{\psi}(s) \tag{4.74}$$

beschrieben. Da im betrachteten LQ-Fall nach Problem 4.3 die optimale Value Function durch $V^*(\tilde{\boldsymbol{x}}) = \tilde{\boldsymbol{x}}^{\mathsf{T}}\tilde{\boldsymbol{P}}^*\tilde{\boldsymbol{x}}$ gegeben ist (vgl. [Föl16, S. 342]), lassen sich wegen[83]

$$V^*(\tilde{\boldsymbol{x}}) = \tilde{\boldsymbol{x}}^{\mathsf{T}}\tilde{\boldsymbol{P}}^*\tilde{\boldsymbol{x}} = \underbrace{(\tilde{\boldsymbol{x}} \otimes_{\mathrm{r}} \tilde{\boldsymbol{x}})^{\mathsf{T}}}_{=:\boldsymbol{\phi}(\tilde{\boldsymbol{x}})}\underbrace{\mathrm{vecr}\left(\tilde{\boldsymbol{P}}^*\right)}_{=:\boldsymbol{w}_{\phi}^*} \tag{4.75}$$

und

$$\begin{aligned}\big(\nabla_{\tilde{\boldsymbol{x}}} V^*(\tilde{\boldsymbol{x}})\big)^{\mathsf{T}}\frac{\mathrm{d}\tilde{\boldsymbol{x}}(t)}{\mathrm{d}t} &= 2\tilde{\boldsymbol{x}}^{\mathsf{T}}\tilde{\boldsymbol{P}}^*\left(\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{B}}\boldsymbol{u}\right)\\ &= (\tilde{\boldsymbol{x}} \otimes_{\mathrm{r}} \tilde{\boldsymbol{x}})^{\mathsf{T}}\,\mathrm{vecr}\left(\tilde{\boldsymbol{A}}^{\mathsf{T}}\tilde{\boldsymbol{P}}^* + \tilde{\boldsymbol{P}}^*\tilde{\boldsymbol{A}}\right) + (\tilde{\boldsymbol{x}} \otimes \boldsymbol{u})^{\mathsf{T}}\,2\mathrm{vec}\left(\tilde{\boldsymbol{B}}^{\mathsf{T}}\tilde{\boldsymbol{P}}^*\right)\\ &= \underbrace{\left[(\tilde{\boldsymbol{x}} \otimes_{\mathrm{r}} \tilde{\boldsymbol{x}})^{\mathsf{T}} \quad (\tilde{\boldsymbol{x}} \otimes \boldsymbol{u})^{\mathsf{T}}\right]}_{=:\boldsymbol{\psi}^{\mathsf{T}}(\tilde{\boldsymbol{x}},\boldsymbol{u})}\underbrace{\begin{bmatrix}\mathrm{vecr}\left(\tilde{\boldsymbol{A}}^{\mathsf{T}}\tilde{\boldsymbol{P}}^* + \tilde{\boldsymbol{P}}^*\tilde{\boldsymbol{A}}\right)\\ 2\mathrm{vec}\left(\tilde{\boldsymbol{B}}^{\mathsf{T}}\tilde{\boldsymbol{P}}^*\right)\end{bmatrix}}_{=:\boldsymbol{w}_{\psi}^*}\end{aligned} \tag{4.76}$$

geeignete Funktionsapproximatoren $\boldsymbol{\phi}(\tilde{\boldsymbol{x}})$ und $\boldsymbol{\psi}(\tilde{\boldsymbol{x}},\boldsymbol{u})$ wählen. Aus der Minimierung der Hamilton-Funktion (vgl. (2.14), (2.18) und (4.72)) sowie der Linearität von $\boldsymbol{\psi}(\tilde{\boldsymbol{x}},\boldsymbol{u})$ bezüglich $\boldsymbol{u}$ folgt

$$\begin{aligned}\boldsymbol{\mu}^*(\tilde{\boldsymbol{x}}) &= \arg\min_{\boldsymbol{u}}\left(\big(\nabla_{\tilde{\boldsymbol{x}}} V^*(\tilde{\boldsymbol{x}})\big)^{\mathsf{T}}\frac{\mathrm{d}\tilde{\boldsymbol{x}}(t)}{\mathrm{d}t} + r(\tilde{\boldsymbol{x}},\boldsymbol{u})\right)\\ &= \arg\min_{\boldsymbol{u}}\big(\boldsymbol{\psi}^{\mathsf{T}}(\tilde{\boldsymbol{x}},\boldsymbol{u})\boldsymbol{w}_{\psi}^* + r(\tilde{\boldsymbol{x}},\boldsymbol{u})\big)\\ &= -\frac{1}{2}\boldsymbol{R}^{\mathsf{T}}\big(\nabla_{\boldsymbol{u}}\boldsymbol{\psi}(\tilde{\boldsymbol{x}},\boldsymbol{u})\big)^{\mathsf{T}}\boldsymbol{w}_{\psi}^*\\ &= -\tilde{\boldsymbol{K}}^*\tilde{\boldsymbol{x}}\end{aligned} \tag{4.77}$$

[83] Hierbei bezeichnet $\otimes$ das Kronecker-Produkt, $\otimes_{\mathrm{r}}$ das reduzierte Kronecker-Produkt, welches nur nicht-redundante Elemente beinhaltet, $\mathrm{vec}(\cdot)$ die Vektorisierung einer Matrix, wobei die Spalten vertikal konkateniert werden, und $\mathrm{vecr}(\cdot)$ die Vektorisierung einer symmetrischen Matrix $\boldsymbol{M}$, sodass $\boldsymbol{x}^{\mathsf{T}}\boldsymbol{M}\boldsymbol{x} = (\boldsymbol{x} \otimes_{\mathrm{r}} \boldsymbol{x})^{\mathsf{T}}\mathrm{vecr}(\boldsymbol{M})$ gilt.

mit

$$\tilde{\boldsymbol{K}}^* = \frac{1}{2}\boldsymbol{R}^{-1}\mathrm{mat}\left(\left[\boldsymbol{w}_\psi^*\right]_{\frac{\tilde{n}(\tilde{n}+1)}{2}:\left(\frac{\tilde{n}(\tilde{n}+1)}{2}+\tilde{n}p\right)}, p, \tilde{n}\right), \tag{4.78}$$

wobei $\tilde{n} = n + n_\zeta$ der Dimension des erweiterten Zustands $\tilde{\boldsymbol{x}}$ und p der Dimension der Eingangsgröße $\boldsymbol{u}$ entspricht[84]. Damit reduziert sich das ADP-Problem auf das datenbasierte Bestimmen insbesondere des Gewichts $\boldsymbol{w}_\psi^*$. Das Vorgehen der hierbei verwendeten Value Iteration [BJ16a] ist in Anhang B.8 skizziert.

Da eine ausreichende Systemanregung, d. h. Regularität der Matrizen $\boldsymbol{D}_\phi$ (B.34) und $\boldsymbol{D}_\psi$ (B.35), während der Datenaufzeichnung für einen erfolgreichen Adaptionsprozess benötigt wird [BJ16a, Assumption 3] (vgl. auch Kapitel 5), wird Gaußsches weißes Rauschen mit der Standardabweichung σ_{expl} zum Systemeingang $\boldsymbol{u}(t)$ addiert. Weil hierdurch jedoch nur der Systemzustand $\boldsymbol{x}(t)$ und nicht der erweiterte Zustand $\tilde{\boldsymbol{x}}(t) = \begin{bmatrix}\boldsymbol{x}^\mathsf{T}(t) & \boldsymbol{\zeta}^\mathsf{T}(t)\end{bmatrix}^\mathsf{T}$ angeregt wird, muss zudem der Zustand $\boldsymbol{\zeta}(t)$, der den Solltrajektorienverlauf repräsentiert, angeregt werden. Um angeregte Daten $\boldsymbol{\zeta}(t)$ für den Adaptionsprozess zu generieren, ohne dabei den Systemzustand $\boldsymbol{x}(t)$ weiter zu beeinflussen, wird die in Abbildung 4.3 gezeigte Struktur verwendet.

Der Referenzzustand $\boldsymbol{\zeta}_{\text{train}}(t)$ wird zu Beginn jedes Integrationsintervalls der Länge T_{IRL} zufällig initialisiert, indem für jedes Element ein zufälliger Wert aus der Normalverteilung $\mathcal{N}\left(0; \sigma_{\text{expl}}^2\right)$ gezogen wird. Hierdurch wird einerseits sichergestellt, dass unterschiedliche Referenzzustände in den Trainingsdaten vorhanden sind, andererseits folgt $\boldsymbol{\zeta}_{\text{train}}$ innerhalb eines IRL-Integrationsintervalls der Dauer T_{IRL} der durch $\boldsymbol{D}' = \boldsymbol{D} - \frac{\gamma}{2}\boldsymbol{I}$ beschriebenen Dynamik, weshalb eine ADP-kompatible Solltrajektoriendarstellung (vgl. Definition 4.1) zur

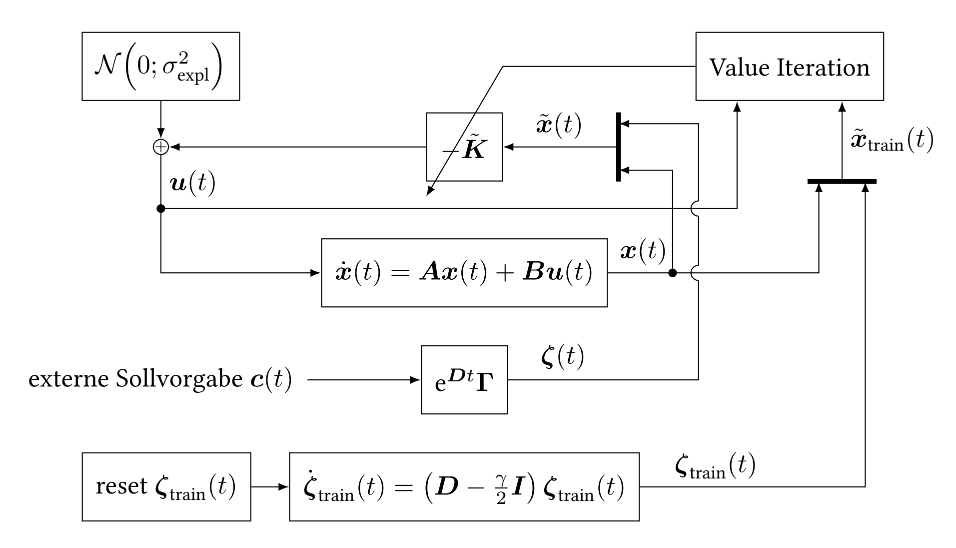


Abbildung 4.3: Struktur des vorgestellten zeitkontinuierlichen ADP-Trajektorienfolgeregelungsansatzes.

[84] Der Operator mat$(\cdot)$ bildet aus einem Vektor eine Matrix, wobei $\mathrm{mat}(\mathrm{vec}(\boldsymbol{M}), \tilde{n}, p) = \boldsymbol{M} \in \mathbb{R}^{\tilde{n}\times p}$ gilt, zudem bezeichne $\left[\boldsymbol{v}\right]_{n_1:n_2} = \begin{bmatrix}v_{n_1} & \dots & v_{n_2}\end{bmatrix}^\mathsf{T}$ mit $n_1, n_2 \in \mathbb{N}_{>0}, n_1 < n_2$, einen Teilvektor eines Vektors $\boldsymbol{v}$.

Erzeugung der Lerndaten vorliegt. Unabhängig davon kann der gewünschte Verlauf der Trajektorie durch beliebige Vorgabe des Parameters $\boldsymbol{c}(t)$ erfolgen.

4.2.5 Simulationsergebnisse

Die simulative Evaluation des zuvor vorgestellten ADP-basierten Solltrajektorienfolgereglers erfolgt anhand einer Implementierung in MATLAB R2020a (64 Bit) und SIMULINK 10.1. Zunächst werden das während der Simulation verwendete Beispielsystem und die Trainingsparameter vorgestellt. Danach werden durch Wahl der Hyperparameter λ_{ij} und ν_{ij} verschiedene Solltrajektorienparametrierungen betrachtet (vgl. Bemerkung 4.1) und mit der aus der Literatur bekannten stationären Sollwertvorgabe verglichen. Abschließend wird der Einfluss von Messrauschen, das auf den Systemzustand $\boldsymbol{x}(t)$ addiert wird, simulativ untersucht.

4.2.5.1 Systemmodell und Parametrierung des ADP-Algorithmus

Betrachtet wird im Folgenden ein Feder-Masse-Dämpfer-System mit der Masse $m_{\text{sys}} = 1\,\text{kg}$, der Federsteifigkeit $k_{\text{sys}} = 1\,\text{N}\,\text{m}^{-1}$ und der Dämpfung $d_{\text{sys}} = 1\,\text{N}\,\text{s}\,\text{m}^{-1}$, auf das eine Kraft als Stellgröße wirkt. Das System ist demnach durch die Zustandsdifferenzialgleichung

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 0 & 1 \\ -\frac{k_{\text{sys}}}{m_{\text{sys}}} & -\frac{d_{\text{sys}}}{m_{\text{sys}}} \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0 \\ \frac{1}{m_{\text{sys}}} \end{bmatrix} u(t) = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \tag{4.79}$$

beschrieben. Zudem sei $\boldsymbol{R} = 1$, $\boldsymbol{Q} = \begin{bmatrix} 100 & 0 \\ 0 & 0 \end{bmatrix}$ und $\gamma = 0{,}01$ (vgl. Problem 4.3). Somit soll der Zustand $x_1(t)$ einer noch näher zu spezifizierenden Solltrajektorie folgen. Der Initialzustand werde zu $\boldsymbol{x}(0) = \begin{bmatrix} 1 & 1 \end{bmatrix}^{\mathsf{T}}$ gesetzt.

Für den Trainingsvorgang wird das Integrationsintervall zu $T_{\text{IRL}} = 0{,}01\,\text{s}$, das Anregungsrauschen zu $\sigma_{\text{expl}} = 1$ und die Anzahl der verwendeten Tupel zu $M = 200$ gesetzt. Die zu lernenden Gewichte werden mit $\hat{\boldsymbol{w}}_\psi(0) = \boldsymbol{0}$ und $\hat{\boldsymbol{w}}_\phi(0) = \text{vecr}(\boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z})$ initialisiert, wobei $\boldsymbol{Z} \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$ zufällige Elemente[85] enthält, wodurch die initiale Value Function positiv definit ist (vgl. [BJ16a, Theorem 2]).

4.2.5.2 Stationäre Sollwertvorgabe (Vergleichsmethode)

Der vorgestellte Ansatz erlaubt als Spezialfall auch die in der Literatur häufig verwendete stationäre Sollwertvorgabe (vgl. Abschnitt 2.2). Dieser Fall, welcher der Vorgabe eines Polynoms nullter Ordnung entspricht, soll zu Vergleichszwecken hinzugezogen werden. Mit $J_1 = 1$, $\lambda_{11} = 0$ und $\nu_{11} = 1$ folgt $\boldsymbol{D} = 0$, $\boldsymbol{S} = \begin{bmatrix} 1 & 0 \end{bmatrix}^{\mathsf{T}}$ und

$$\tilde{\boldsymbol{K}}^* = \begin{bmatrix} 9{,}0499 & 3{,}3703 & -9{,}9288 \end{bmatrix}. \tag{4.80}$$

[85] Konkret wird eine Gleichverteilung im Intervall $(0; 0{,}1]$ verwendet.

Der Parameter $c(t)$ entspricht in diesem Fall dem Solltrajektorienverlauf $x_{\mathrm{r},1}(t)$. Abbildung 4.4 zeigt den Verlauf der Reglerparameter $\tilde{\boldsymbol{K}}(t)$ und die Gewichtsfehlernorm $\left\|\tilde{\boldsymbol{K}}(t) - \tilde{\boldsymbol{K}}^*\right\|_2$. Die ohne Verwendung des Systemmodells gelernten Reglergewichte konvergieren bis auf eine Fehlernorm von $6{,}28 \cdot 10^{-14}$.

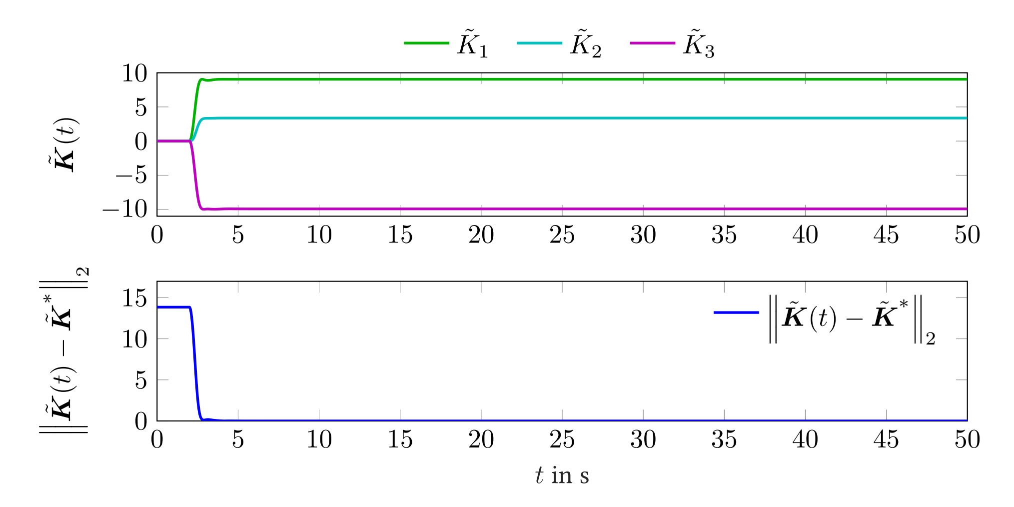


Abbildung 4.4: Reglergewichte $\tilde{\boldsymbol{K}}(t)$ (oben) und Gewichtsfehlernorm $\left\|\tilde{\boldsymbol{K}}(t) - \tilde{\boldsymbol{K}}^*\right\|_2$ (unten) für das Beispiel der stationären Sollvorgabe. Während der ersten 2 s werden $M = 200$ Datentupel aufgezeichnet.

4.2.5.3 Harmonischer Oszillator als Solltrajektorienparametrierung

In diesem Beispiel werden zur Beschreibung von $x_{\mathrm{r},1}$ zwei konjugiert-komplexe Polpaare $\lambda_{11} = \lambda_{12}^\star = \mathrm{j}\omega_1$ und $\lambda_{13} = \lambda_{14}^\star = \mathrm{j}\omega_2$ mit $\nu_{1j} = 1$, $j = 1, \ldots, 4$, verwendet (vgl. Annahme 4.1). Während die Hyperparameter durch die beispielhaft gewählten Werte $\omega_1 = 0{,}5$ und $\omega_2 = 1{,}3$ die Frequenzen der darstellbaren harmonischen Schwingungen festlegen, lassen sich durch $c_{110} = c_{120}^\star \in \mathbb{C}$ und $c_{130} = c_{140}^\star \in \mathbb{C}$ nach Bemerkung 4.1 deren Phasenverschiebungen und Amplituden parametrieren. Da die Konstruktion nach (4.14) zu einem komplexwertigen $\boldsymbol{D}$ führt, wird die in Anhang B.3 gezeigte Zustandstransformation angewandt. Dies führt auf

$$\boldsymbol{D} = \begin{bmatrix} 0 & \omega_1 & 0 & 0 \\ -\omega_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \omega_2 \\ 0 & 0 & -\omega_2 & 0 \end{bmatrix} \quad \text{und} \quad \boldsymbol{S} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.81}$$

Das optimale Regelgesetz berechnet sich nach Satz 4.1 zu

$$\tilde{\boldsymbol{K}}^* = \begin{bmatrix} 9{,}0499 & 3{,}3703 & -9{,}6992 & -2{,}1628 & -8{,}1630 & -5{,}5458 \end{bmatrix}. \tag{4.82}$$

Die Sollzustandstrajektorie wird durch den Parameter $\boldsymbol{c}$ beschrieben. Dieser wird zu den in Tabelle 4.1 gezeigten Zeitpunkten auf die dort aufgeführten Werte gesetzt und anschließend konstant gehalten.

Zeit t	$c_{110} = c_{120}^{\star}$	$2\lvert c_{110}\rvert$	$c_{130} = c_{140}^{\star}$	$2\lvert c_{130}\rvert$
$0\,\mathrm{s}$	$0{,}1768 + \mathrm{j}0{,}1768$	$0{,}5$	$0{,}2500 + \mathrm{j}0$	$0{,}5$
$9{,}7\,\mathrm{s}$	$0 + \mathrm{j}0$	0	$0{,}1294 + \mathrm{j}0{,}4830$	1
$18{,}7\,\mathrm{s}$	$0{,}2000 + \mathrm{j}0$	$0{,}4$	$0 + \mathrm{j}0{,}1000$	$0{,}2$

Tabelle 4.1: Wahl des Parameters $\boldsymbol{c}$ für das Beispiel der harmonischen Referenzzustandsdarstellung. Nach (4.11) stellt $2|c_{110}|$ die Amplitude der Schwingung mit Frequenz $\omega_1 = 0{,}5$ und $2|c_{130}|$ die Amplitude zur Frequenz $\omega_2 = 1{,}3$ dar.

Der resultierende Sollzustandsverlauf $x_{\mathrm{r},1}(t)$, Zustandsverlauf $\boldsymbol{x}(t)$ und Stellgrößenverlauf $u(t)$ ist in Abbildung 4.5 gezeigt. Während der ersten zwei Sekunden wird der Datenspeicher mit $M = 200$ Tupeln gefüllt. Danach werden die Gewichte $\hat{\boldsymbol{w}}_{\phi}$ und $\hat{\boldsymbol{w}}_{\psi}$ adaptiert und das Anregungsrauschen zu null gesetzt. Der Zustand x_1 folgt dem Sollzustandsverlauf $x_{\mathrm{r},1}$ optimal im Sinne des Gütemaßes (4.46). Dies bestätigt Abbildung 4.6, in welcher der zeitliche Verlauf der gelernten Reglerparameter $\tilde{\boldsymbol{K}}(t)$ sowie deren Gewichtsfehlernorm $\left\|\tilde{\boldsymbol{K}}(t) - \tilde{\boldsymbol{K}}^{*}\right\|_2$ zu sehen sind. Die Reglergewichte konvergieren hierbei bis auf eine Fehlernorm von $2{,}83 \cdot 10^{-13}$. Der Rang der Datenmatrizen $\boldsymbol{D}_{\phi}$ und $\boldsymbol{D}_{\psi}$ (vgl. Anhang B.8) ist in Abbildung 4.7 gezeigt, welche eine ausreichende Anregung der Daten bestätigt.

In Abbildung 4.5 sind des Weiteren die Trajektorien $x_{1,\mathrm{s}}(t)$, $x_{2,\mathrm{s}}(t)$ und $u_{\mathrm{s}}(t)$ der Vergleichslösung mit stationärer Sollzustandsvorgabe nach Abschnitt 4.2.5.2 gezeigt. Da die Vorgabe des aktuellen Sollzustands $x_{\mathrm{r},1}(t)$ keinerlei Information über den weiteren Verlauf der Solltrajektorie beinhaltet, nimmt der Vergleichsregler mit stationärer Sollwertvorgabe stets an, der aktuelle Sollzustand solle bis ins Unendliche gehalten werden. Dies führt zu einem merklichen Zeitversatz der Trajektorie $x_{1,\mathrm{s}}(t)$. Wie Abbildung 4.8 verdeutlicht, korreliert dieser Zeitversatz der Trajektorie mit deutlich höheren Kosten. Die in dieser Arbeit präsentierte Methode berücksichtigt hingegen den zukünftigen Verlauf der Solltrajektorie und führt aufgrund der damit verbundenen Vorausschaufähigkeit zu signifikant geringeren Kosten im Sinne des Gütefunktionals. So werden die kumulierten Kosten am Ende des Simulationsbeispiels, wie in Abbildung 4.8 ersichtlich, um etwa 69 % reduziert.

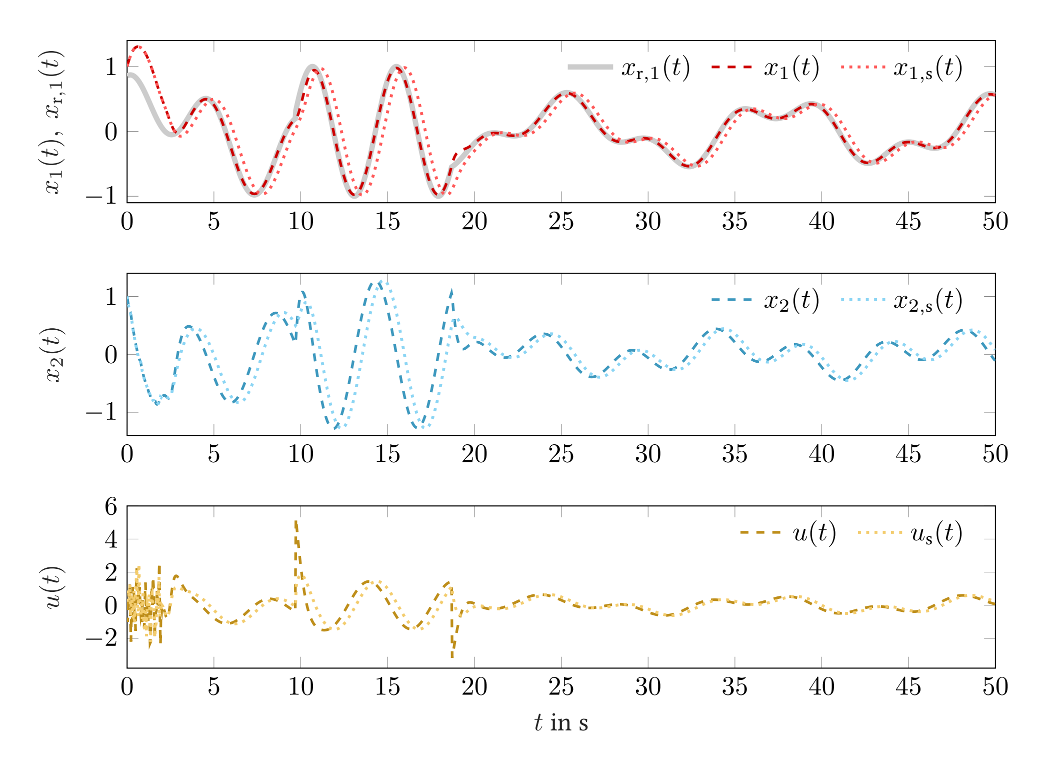


Abbildung 4.5: Sollzustandsverlauf $x_{\mathrm{r},1}(t)$, Zustandsverlauf $\boldsymbol{x}(t)$ und Stellgröße $u(t)$ für das Beispiel des harmonischen Oszillators als Solltrajektorienparametrierung im Vergleich zur stationären Sollzustandsvorgabe (gekennzeichnet durch den Index s).

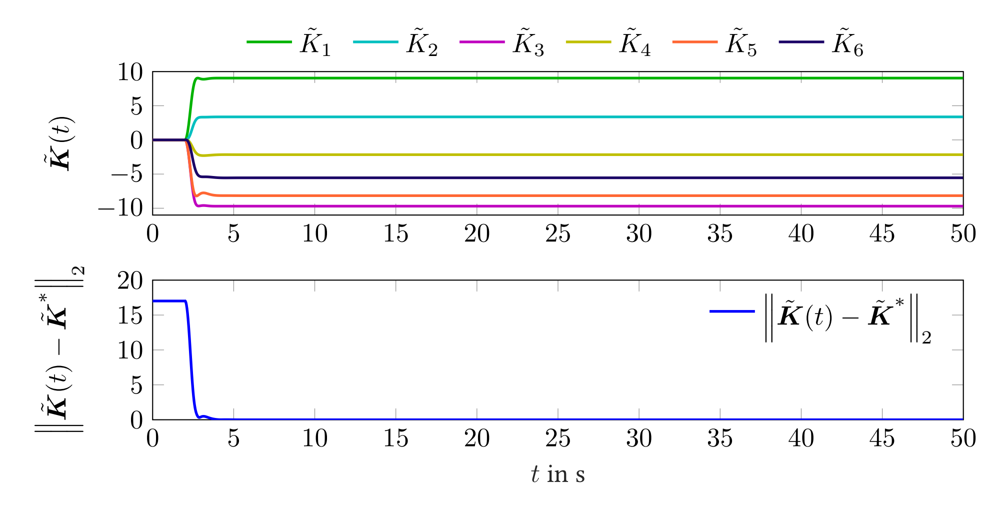


Abbildung 4.6: Reglergewichte $\tilde{\boldsymbol{K}}(t)$ (oben) und Gewichtsfehlernorm $\left\|\tilde{\boldsymbol{K}}(t) - \tilde{\boldsymbol{K}}^*\right\|_2$ (unten) für das Beispiel des harmonischen Oszillators als Solltrajektorienparametrierung. Während der ersten 2 s werden $M = 200$ Datentupel aufgezeichnet.

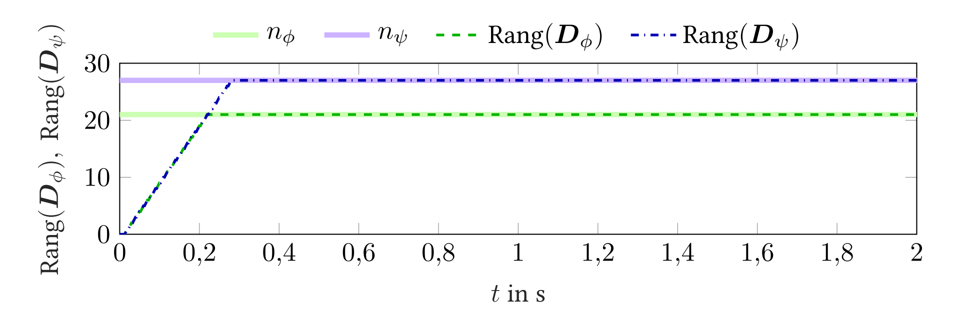


Abbildung 4.7: Rang der Datenmatrizen $\boldsymbol{D}_\phi \in \mathbb{R}^{n_\phi \times n_\phi}$ und $\boldsymbol{D}_\psi \in \mathbb{R}^{n_\psi \times n_\psi}$ während der ersten 2 s. Für das gewählte Beispiel des harmonischen Oszillators als Solltrajektorienparametrierung gilt $n_\phi = \frac{\tilde{n}(\tilde{n}+1)}{2} = 21$ und $n_\psi = \frac{\tilde{n}(\tilde{n}+1)}{2} + \tilde{n}p = 27$.

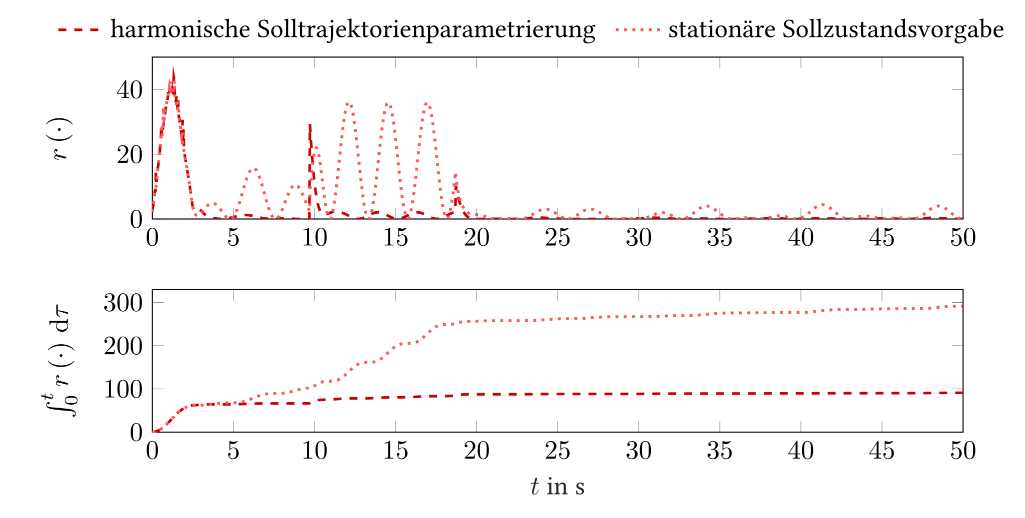


Abbildung 4.8: Kosten $r(\boldsymbol{x}(t), \boldsymbol{x}_\mathrm{r}(t), \boldsymbol{u}(t))$ und kumulierte Kosten $\int_0^t r(\boldsymbol{x}(\tau), \boldsymbol{x}_\mathrm{r}(\tau), \boldsymbol{u}(\tau))\,\mathrm{d}\tau$ für das Beispiel des harmonischen Oszillators als Solltrajektorienparametrierung im Vergleich zur stationären Sollzustandsvorgabe.

4.2.5.4 Polynomvorgabe

Im Folgenden wird zur Beschreibung des Sollverlaufs des ersten Systemzustands ein Polynom zweiten Grades verwendet, d. h. es gilt

$$x_{\mathrm{r},1}(t) = c_{110} + c_{111}t + c_{112}t^2. \tag{4.83}$$

Dies entspricht nach Bemerkung 4.1 einem dreifachen Eigenwert bei null ($\lambda_{11} = 0$, $\nu_{11} = 3$). Mit

$$\boldsymbol{D} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{und} \quad \boldsymbol{S} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.84}$$

berechnet sich die optimale Zustandsrückführungsmatrix zu

$$\tilde{\boldsymbol{K}}^* = \begin{bmatrix} 9{,}0499 & 3{,}3703 & -9{,}9288 & -4{,}3181 & -0{,}8922 \end{bmatrix}. \tag{4.85}$$

Der Parameter $\boldsymbol{c}$, der den Solltrajektorienverlauf beschreibt, wird, wie in Tabelle 4.2 gezeigt, gesetzt und zwischen den dort aufgeführten Zeitpunkten konstant gehalten.

Zeit t	$\boldsymbol{x}_{\mathrm{r},1}(t)$	$\dot{\boldsymbol{x}}_{\mathrm{r},1}(t)$	$\ddot{\boldsymbol{x}}_{\mathrm{r},1}(t)$	c_{110}	c_{111}	c_{112}
0 s	0	0	0	0	0	0
5 s	0	1	0	−5	1	0
10 s	5	0	0	5	0	0
15 s	5	0	−0,6	−62,5	9	−0,3
18 s	2,3	−1,4	1	189,5	−19,4	0,5
20 s	1,5	1	−0,5	−118,5	11	−0,25
25 s	0,25	−2	1,4	487,75	−37	0,7
26,4 s	−1,1780	0	0	−1,1780	0	0
28 s	−1,1780	0	0,07	26,2620	−1,96	0,035
35 s	0,5370	0,5	−0,1	−78,2130	4	−0,05

Tabelle 4.2: Wahl des Parameters $\boldsymbol{c}$ für das Beispiel der Polynomvorgabe. Hierbei entspricht $\boldsymbol{\zeta}(t) = \begin{bmatrix} \boldsymbol{x}_{\mathrm{r},1}(t) & \dot{\boldsymbol{x}}_{\mathrm{r},1}(t) & \ddot{\boldsymbol{x}}_{\mathrm{r},1}(t) \end{bmatrix}^\mathsf{T}$ (vgl. (4.12)).

Der Sollzustandsverlauf $x_{\mathrm{r},1}(t)$, Zustandsverlauf $\boldsymbol{x}(t)$ und Stellgrößenverlauf $u(t)$ ist in Abbildung 4.9 gegeben. Abbildung 4.10 zeigt den zeitlichen Verlauf der geschätzten Reglerparameter $\tilde{\boldsymbol{K}}(t)$ und die Gewichtsfehlernorm $\left\| \tilde{\boldsymbol{K}}(t) - \tilde{\boldsymbol{K}}^* \right\|_2$. Die Reglergewichte konvergieren bis auf eine Fehlernorm von $2{,}63 \cdot 10^{-8}$. Der Rang der Datenmatrizen $\boldsymbol{D}_\phi$ und $\boldsymbol{D}_\psi$ ist Abbildung 4.11 zu entnehmen.

Auch in diesem Fall weist der in dieser Arbeit vorgestellte Ansatz, eine ADP-kompatible Beschreibungsform des Solltrajektorienverlaufs zu verwenden, Vorteile gegenüber der Vergleichsmethode nach Abschnitt 4.2.5.2 auf. So folgt gemäß Abbildung 4.9 der Zustand $x_1(t)$ dem Sollzustand $x_{\mathrm{r},1}(t)$ präziser im Vergleich zu $x_{1,\mathrm{s}}(t)$. Letztlich kann, wie Abbildung 4.12 offenbart, der neuartige Ansatz auch in diesem Beispiel die Kosten im Vergleich zur stationären Sollzustandsvorgabe signifikant reduzieren. Die kumulierten Kosten werden in dieser beispielhaften Simulation um 38 % verringert.

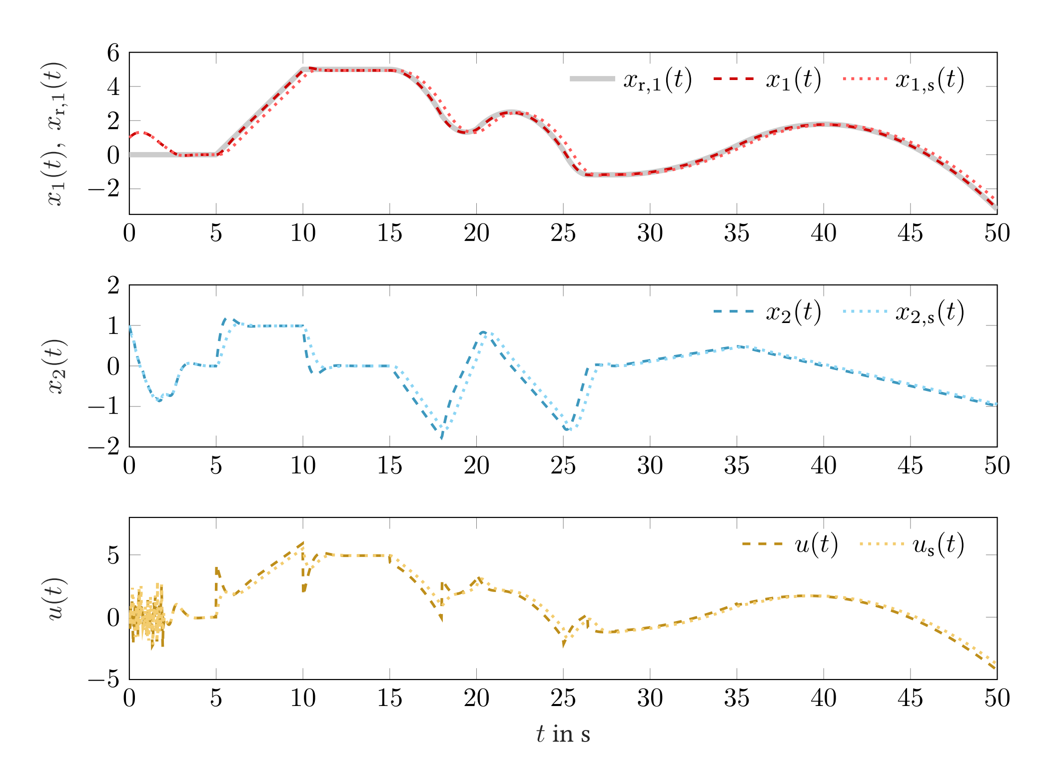


Abbildung 4.9: Sollzustandsverlauf $x_{\mathrm{r},1}(t)$, Zustandsverlauf $\boldsymbol{x}(t)$ und Stellgröße $u(t)$ für das Beispiel der Polynomvorgabe im Vergleich zur stationären Sollzustandsvorgabe (gekennzeichnet durch den Index s).

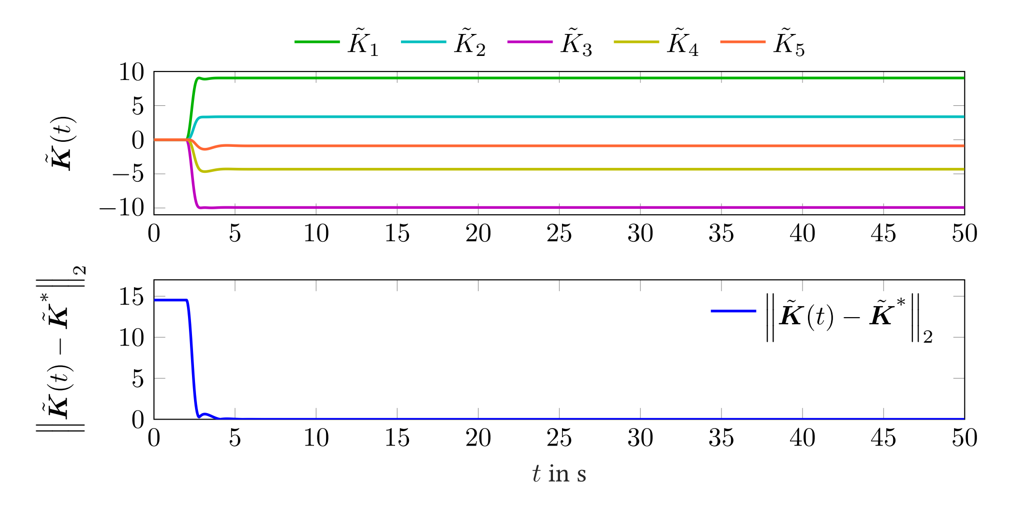


Abbildung 4.10: Reglergewichte $\tilde{\boldsymbol{K}}(t)$ (oben) und Gewichtsfehlernorm $\left\|\tilde{\boldsymbol{K}}(t) - \tilde{\boldsymbol{K}}^*\right\|_2$ (unten) für das Beispiel der Polynomvorgabe. Während der ersten 2 s werden $M = 200$ Datentupel aufgezeichnet.

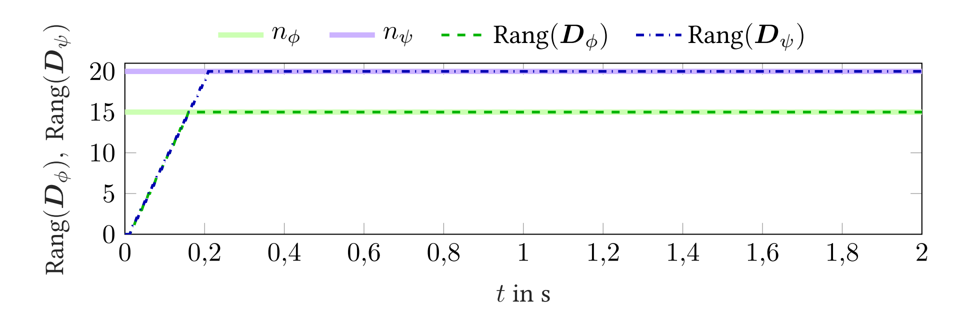


Abbildung 4.11: Rang der Datenmatrizen $\boldsymbol{D}_\phi \in \mathbb{R}^{n_\phi \times n_\phi}$ und $\boldsymbol{D}_\psi \in \mathbb{R}^{n_\psi \times n_\psi}$ während der ersten 2 s. Für das gewählte Beispiel der Polynomvorgabe gilt $n_\phi = \frac{\tilde{n}(\tilde{n}+1)}{2} = 15$ und $n_\psi = \frac{\tilde{n}(\tilde{n}+1)}{2} + \tilde{n}p = 20$.

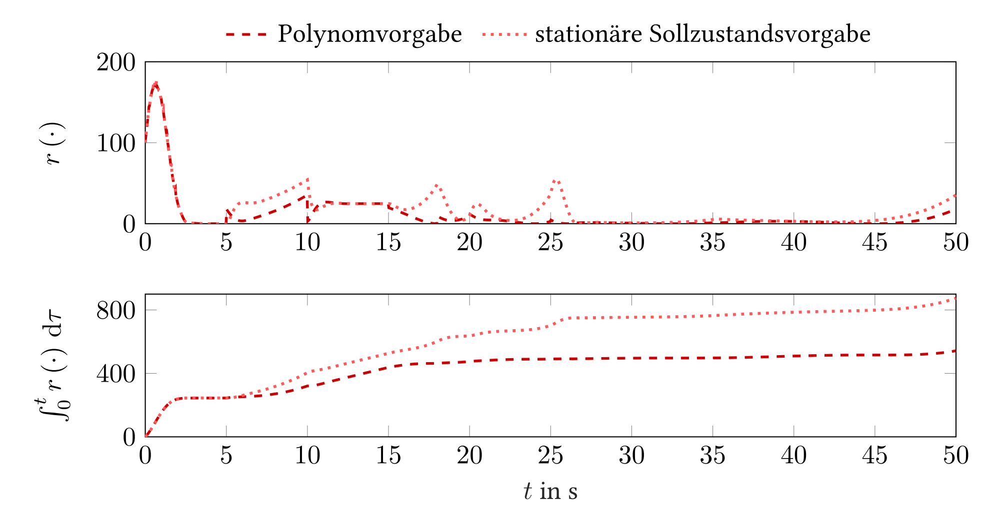


Abbildung 4.12: Kosten $r(\boldsymbol{x}(t), \boldsymbol{x}_\mathrm{r}(t), \boldsymbol{u}(t))$ und kumulierte Kosten $\int_0^t r(\boldsymbol{x}(\tau), \boldsymbol{x}_\mathrm{r}(\tau), \boldsymbol{u}(\tau))\, \mathrm{d}\tau$ für das Beispiel der Polynomvorgabe im Vergleich zur stationären Sollzustandsvorgabe.

4.2.5.5 Einfluss von Messrauschen

Abschließend wird für das gegebene Simulationsbeispiel noch der Einfluss von weißem Gaußschem Messrauschen mit unterschiedlicher Standardabweichung σ_m, das zu $\boldsymbol{x}(t)$ addiert wird, untersucht. Hierbei wird die zuvor verwendete Solltrajektoriendarstellung der Polynomvorgabe wie in Abschnitt 4.2.5.4 verwendet. Abbildung 4.13 zeigt den Zustandsverlauf $x_1(t)$ sowie die Stellgröße $u(t)$ für Standardabweichungen des Messrauschens von $\sigma_\mathrm{m} = 0$ bis $\sigma_\mathrm{m} = 2$. Ein Messrauschen bis $\sigma_\mathrm{m} = 0{,}2$ führt zu kaum merklichen Abweichungen. Bis zu einem Messrauschen von $\sigma_\mathrm{m} = 1$ bleibt der geschlossene Regelkreis stabil und kann der vorgegebenen Solltrajektorie folgen. Für $\sigma_\mathrm{m} = 2$ resultiert ein instabiles Regelgesetz. Die Fehlernorm im Vergleich zum optimalen Regelgesetz ist für unterschiedliches Messrauschen in Abbildung 4.14

gegeben. Dabei ist zu erkennen, dass der gelernte Regler mit zunehmendem Messrauschen stärker vom Optimalregler abweicht. Für $\sigma_\mathrm{m} = 2$ divergiert $\tilde{\boldsymbol{K}}$ für das Beispielszenario.

Zwar weisen in allen betrachteten Fällen die Datenmatrizen $\boldsymbol{D}_\phi$ und $\boldsymbol{D}_\psi$ vollen Rang auf und das Messrauschen begünstigt diese Tatsache sogar, jedoch ist das Systemverhalten bei zu starkem Messrauschen im Vergleich zur Systemanregung nicht mehr adäquat in den Messdaten repräsentiert[86]. Somit treten auch hier vergleichbare Effekte wie im zeitdiskreten Fall in Abschnitt 3.3.4.4 auf und die Notwendigkeit einer geeigneten Systemanregung verstärkt sich insbesondere im Fall vorhandenen Messrauschens.

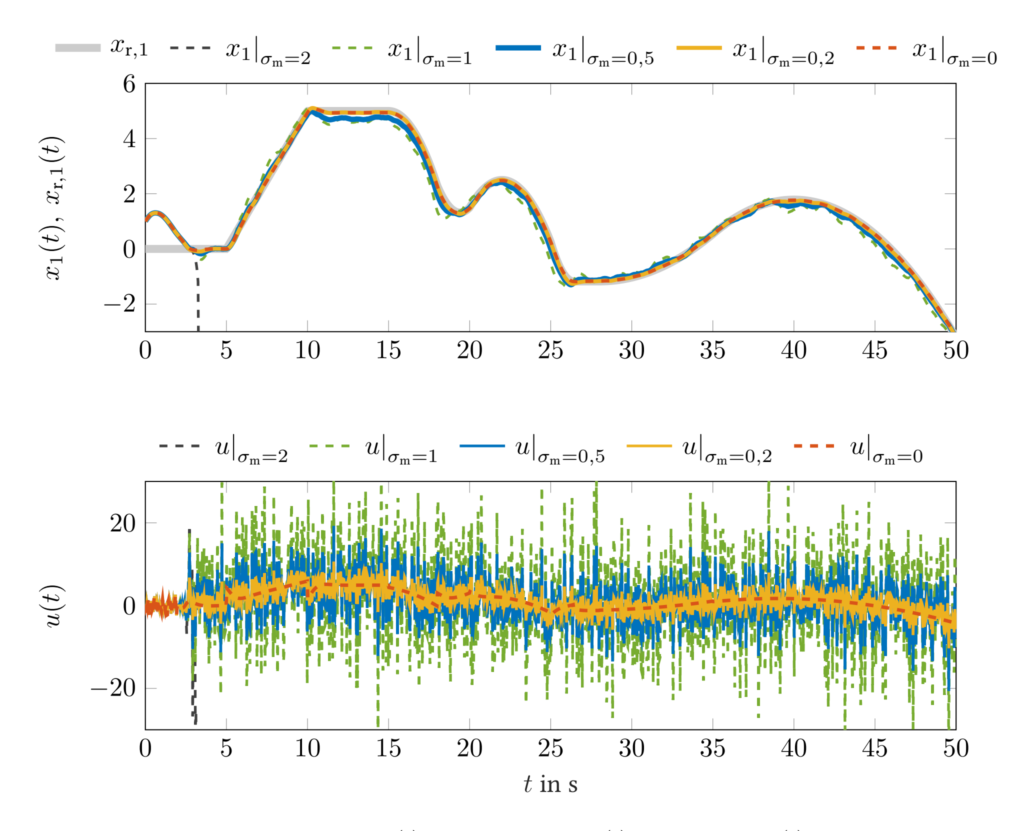


Abbildung 4.13: Sollzustandsverlauf $x_{\mathrm{r},1}(t)$, Zustandsverlauf $\boldsymbol{x}_1(t)$ und Stellgröße $u(t)$ für das Beispiel der Polynomvorgabe unter Einfluss von Messrauschen.

[86] Um stärkeres additives Messrauschen handhabbar zu machen, kann hingegen (im Rahmen der numerischen Grenzen) mit einer größeren Systemauslenkung während der Datenaufzeichnung gearbeitet werden, d. h. beispielsweise mit einem stärkeren Anregungsrauschen. Für $\sigma_\mathrm{expl} = 30$ und ein durch $\sigma_\mathrm{m} = 2$ beschriebenes Messrauschen ergibt sich die Fehlernorm des gelernten Regelgesetzes beispielsweise zu 4,2 und der geschlossene Regelkreis ist stabil. Diese Wahl des Anregungs- und Messrauschens führt zu einer ähnlichen Abweichung vom optimalen Regelgesetz wie zuvor mit $\sigma_\mathrm{expl} = 1$ und $\sigma_\mathrm{m} = 1$ mit einer Fehlernorm von 4,7.

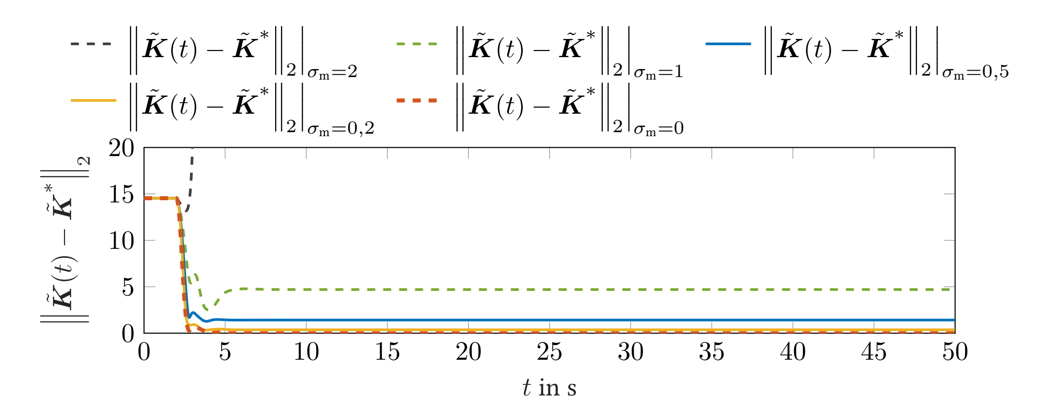


Abbildung 4.14: Gewichtsfehlernorm $\left\| \tilde{\boldsymbol{K}}(t) - \tilde{\boldsymbol{K}}^* \right\|_2$ für das Beispiel der Polynomvorgabe unter Einfluss von Messrauschen. Während der ersten 2 s werden $M = 200$ Datentupel aufgezeichnet.

4.3 Zusammenfassung

In diesem Kapitel wurde erstmals eine allgemeine Definition zeitkontinuierlicher ADP-kompatibler Solltrajektoriendarstellungen vorgestellt. Des Weiteren wurde eine neuartige Klasse ADP-kompatibler Solltrajektorien in Form einer Superposition von Lösungen homogener linearer Differenzialgleichungen präsentiert und analysiert. Diese verallgemeinerte Darstellung inkludiert insbesondere konstante Sollzustandsvorgaben, Polynomverläufe, Exponentialfunktionen und harmonische Funktionen. Für LQ-Solltrajektorienfolgeregelungsprobleme mit gegebenenfalls gedämpfter Solltrajektoriendynamik, die diese neue, ADP-kompatible Solltrajektoriendarstellung nutzen, wurde unter milden Annahmen die Existenz und Eindeutigkeit der stabilisierenden optimalen Lösung gezeigt.

Die Einbeziehung des durch den Parameter $\boldsymbol{c}$ repräsentierten Solltrajektorienverlaufs in die Value Function V ermöglicht schließlich ein ADP-basiertes, modellfreies Erlernen sowohl der optimalen Value Function als auch des zugehörigen Regelgesetzes. Der hierzu verwendete ADP-Algorithmus ist dabei grundsätzlich austauschbar, beispielhaft wurde die IRL-basierte Value Iteration nach Bian und Jiang [BJ16a] genutzt. Der für den Trainingsvorgang verwendete Solltrajektorienparameter $\boldsymbol{\zeta}_{\text{train}}$ folgt hierbei während eines Integrationsintervalls der Dauer T_{IRL} der durch $\boldsymbol{D} - \frac{\gamma}{2}\boldsymbol{I}$ beschriebenen Dynamik und wird nur zu Beginn der neuen Integrationsintervalle zugunsten einer ausreichenden Anregung neu initialisiert. Insbesondere gehören die während des Trainings verwendeten Datentupel $\left(\tilde{\boldsymbol{x}}(t_j), \boldsymbol{\Theta}_j, \tilde{\boldsymbol{x}}(t_j + T_{\text{IRL}})\right), j = 1, \dots, M$, (vgl. Anhang B.8) daher zu ADP-kompatiblen Solltrajektorien (vgl. auch Bemerkung 3.1). Aufgrund der in Abbildung 4.3 gezeigten parallelen Struktur von $\boldsymbol{\zeta}(t)$ und $\boldsymbol{\zeta}_{\text{train}}(t)$ kann der Parameter $\boldsymbol{c}$, der den Solltrajektorienverlauf aus lokaler Perspektive zum Zeitpunkt t beschreibt, jederzeit beliebig vorgegeben werden. Der im vorliegenden Kapitel vorgestellte Ansatz kann letztlich als Übertragung der ADP-kompatiblen zeitdiskreten Methode aus Abschnitt 3.2 auf

zeitkontinuierliche Systeme interpretiert werden[87]. Als Grenze des Ansatzes lässt sich nennen, dass in der Zukunft liegende Sprünge durch die Solltrajektoriendarstellung nach (4.7) nicht exakt abgebildet, sondern lediglich approximiert werden können. Zudem muss der Einfluss der Systemanregung auf die Trainingsdatentupel signifikant stärker sein als der Einfluss des Messrauschens (vgl. Abschnitt 4.2.5.5). Letzteres ist jedoch kein spezifisches Problem der ADP-kompatiblen Solltrajektorienfolgeregelung, sondern Systemidentifikationsproblemen und adaptiven Methoden inhärent (vgl. [Bit84], [KT04, S. 122]).

Die vorgestellten Simulationsergebnisse zeigen die Flexibilität der vorgestellten Methode auf. So lassen sich im Rahmen ADP-kompatibler Solltrajektoriendarstellungen vielfältige Referenztrajektorienverläufe in modellfreie zeitkontinuierliche ADP-Algorithmen integrieren. Die explizite Abhängigkeit der gelernten Value Function und des zugehörigen Optimalreglers vom Parameter $\boldsymbol{c}$ erlaubt im Gegensatz zu bestehenden Ansätzen einerseits, vielfältige Solltrajektorien von außen vorzugeben, andererseits kann hierdurch der zukünftige Verlauf der Solltrajektorie berücksichtigt werden. Insbesondere führt die Einbeziehung des Solltrajektorienverlaufs zu einem vorausschauenden Verhalten, das, im Vergleich zu einer aus der Literatur bekannten konstanten Sollwertvorgabe, mit deutlich reduzierten Gesamtkosten korreliert. Somit beantworten die neuartigen, flexibel einsetzbaren Mechanismen für ADP-basierte Solltrajektorienfolgeregler die in Abschnitt 2.4.1 formulierte Forschungsfrage 1 nach ADP-kompatiblen Solltrajektoriendarstellungen und deren Integration in modellfreie ADP-Ansätze für den zeitkontinuierlichen Fall.

[87] Aufgrund der Stetigkeitsanforderungen nach Definition 4.1 sei an dieser Stelle angemerkt, dass eine direkte Übertragung der in Abschnitt 3.3 vorgestellten zeitdiskreten Repräsentation, welche beliebige Sprünge innerhalb des Vorausschauhorizonts exakt berücksichtigt, auf den zeitkontinuierlichen Fall nicht möglich ist.

5 Konvergenzbedingungen zeitkontinuierlicher adaptiver Optimalregler

Ein zentrales Element ADP-basierter Optimalregler stellt eine geeignete Anregung des Systems dar. Diese ermöglicht adaptiven Optimalreglern, Informationen über das Systemverhalten sowie die damit verbundenen Kosten zu gewinnen und sich kostenoptimal anzupassen (vgl. Abschnitt 2.3, (3.29), (3.115), (B.37) und (B.40)).

Den Ausgangspunkt des vorliegenden Kapitels[88] bildet daher die Analyse der zeitkontinuierlichen PE-Bedingung (2.27) und deren Bedeutung für die Konvergenz eines gradientenbasierten Policy-Evaluation-Schrittes (vgl. Abschnitt 2.1.4). Betrachtet wird hierbei ein zeitkontinuierliches, eingangsaffines Differenzialspiel, das eine Generalisierung des durch (2.10) und (2.11) gegebenen Optimierungsproblems darstellt. Auf dieser Basis wird der zentrale Beitrag dieses Kapitels in Form von hinreichenden Bedingungen an den Systemzustand, die garantieren, dass das zur Adaption benötigte Signal die PE-Bedingung erfüllt, hergeleitet. Dieses Signal folgt bei Verwendung polynomieller Basisfunktionen zur Critic-Funktionsapproximation aus einer nichtlinearen Transformation des Systemzustands. Aufbauend auf diesen theoretischen Erkenntnissen wird schließlich am Beispiel eines exakt zustandslinearisierbaren Systems ein geeignetes Anregungssignal mithilfe eines Vorsteuerungsentwurfs konstruiert, um Simulationen zu ermöglichen. Anhand dieses Simulationsbeispiels werden die vorgestellten Anregungssignale mit einer Anregung durch weißes Rauschen verglichen und Unterschiede diskutiert.

In den Abschnitten 5.1–5.3 erfolgt zunächst eine Einführung und Analyse in die betrachtete Problemstellung, bevor in den Abschnitten 5.4–5.7 die Hauptergebnisse dieses Kapitels präsentiert werden.

5.1 Eingangsaffines Differenzialspiel mit unbekannten Gegenspielern

In diesem Kapitel werden eingangsaffine Differenzialspiele, bei denen die Spieler a priori kein genaues Modell über das Verhalten der jeweils anderen Spieler besitzen, gemäß der nachfolgenden Definition betrachtet. Um die Ziele der N auf ein System einwirkenden Regler

[88] Dieses Kapitel basiert auf einem im Rahmen der vorliegenden Arbeit entstandenen eigenen Beitrag [KKBH23].

formal definieren zu können, wird zudem das Lösungskonzept des Nash-Gleichgewichts gemäß Definition 5.2 herangezogen[89].

Definition 5.1 (Eingangsaffines Nicht-Nullsummen-Differenzialspiel mit unbekannten Gegenspielern)
Ein eingangsaffines Nicht-Nullsummen-Differenzialspiel mit unbekannten Gegenspielern sei charakterisiert durch:

1. *Eine eingangsaffine Systemdynamik*

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \sum_{i=1}^{N} \boldsymbol{g}_i(\boldsymbol{x})\boldsymbol{u}_i =: \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} =: \boldsymbol{f} + \boldsymbol{g}\boldsymbol{u}, \\ \boldsymbol{x}(0) &= \boldsymbol{x}_0, \end{aligned} \tag{5.1}$$

   mit dem Systemzustand $\boldsymbol{x} \in \mathbb{R}^n$, dem Anfangszustand $\boldsymbol{x}_0$ und den Stellgrößen $\boldsymbol{u}_i \in \mathbb{R}^{p_i}$ der N Regler (auch Spieler genannt), $i \in \mathcal{N} = \{1, 2, \dots, N\}$, $N \in \mathbb{N}_{\geq 1}$, $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ und $\boldsymbol{g}_i : \mathbb{R}^n \to \mathbb{R}^{n \times p_i}$. Die Kurzschreibweisen $\boldsymbol{g}(\boldsymbol{x}) := \begin{bmatrix}\boldsymbol{g}_1(\boldsymbol{x}) & \boldsymbol{g}_2(\boldsymbol{x}) & \cdots & \boldsymbol{g}_N(\boldsymbol{x})\end{bmatrix}$, $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^{n \times p}$, $p := \sum_{i=1}^{N} p_i$ und $\boldsymbol{u} := \begin{bmatrix}\boldsymbol{u}_1^\intercal & \boldsymbol{u}_2^\intercal & \cdots & \boldsymbol{u}_N^\intercal\end{bmatrix}^\intercal \in \mathbb{R}^p$ werden zur Vereinfachung der Notation verwendet. Der Initialzeitpunkt t_0 wird o. B. d. A. zu $t_0 = 0$ gesetzt. Zudem seien $\boldsymbol{f}(\boldsymbol{x})$ und $\boldsymbol{g}(\boldsymbol{x})$ Lipschitz-stetig auf der kompakten Menge $\mathcal{X} \subset \mathbb{R}^n$, die den Ursprung enthält. Außerdem gelte $\boldsymbol{f}(\boldsymbol{0}) = \boldsymbol{0}$. Das System sei stabilisierbar auf $\mathcal{X}$.

2. *Die Gütemaße*

$$\begin{aligned} J_i(\boldsymbol{x}_0, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_N) &= \int_0^\infty q_i\big(\boldsymbol{x}(\tau)\big) + \sum_{j=1}^{N} \boldsymbol{\mu}_j^\intercal\big(\boldsymbol{x}(\tau)\big)\boldsymbol{R}_{ij}\boldsymbol{\mu}_j\big(\boldsymbol{x}(\tau)\big)\,\mathrm{d}\tau \\ &=: \int_0^\infty r_i(\boldsymbol{x}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_N)\,\mathrm{d}\tau, \end{aligned} \tag{5.2}$$

   wobei $q_i : \mathbb{R}^n \to \mathbb{R}$ eine positiv definite Funktion (im Sinne von [NA05, S. 53]) darstellt, und $\boldsymbol{R}_{ij} \succeq \boldsymbol{0}, \forall j \in \mathcal{N}, i \neq j$, und $\boldsymbol{R}_{ii} = \boldsymbol{R}_{ii}^\intercal \succ \boldsymbol{0}$ gilt.

3. *Das Ziel jedes Reglers $i \in \mathcal{N}$, ein Regelgesetz $\boldsymbol{\mu}_i^*(\boldsymbol{x})$ zu finden, wobei $\{\boldsymbol{\mu}_1^*, \dots, \boldsymbol{\mu}_N^*\}$ eine Feedback-Nash-Lösung (vgl. Definition 5.2) darstellt und aus Sicht von Regler i*
   a) *die Systemdynamik bekannt ist,*
   b) *Messungen von $\boldsymbol{x}(t)$ und $\boldsymbol{\mu}_j(t), j \in \mathcal{N}$, verfügbar sind*[90]
   c) *und Gütemaße J_j und Regelgesetze $\boldsymbol{\mu}_j(\boldsymbol{x})$ der anderen Spieler, d. h. $\forall j \neq i$, unbekannt sind.*

[89] Da durch $\boldsymbol{\mu}_i$, $i \in \{1, 2, \dots, N\}$, im Folgenden Zustandsrückführungen bezeichnet werden, handelt es sich hierbei um ein sogenanntes Feedback-Nash-Gleichgewicht (vgl. [BO99, Abschnitt 6.5.2]).

[90] Die Notation $\boldsymbol{\mu}_j(t)$ stellt Zeitsignale dar, im Gegensatz zu Regelgesetzen $\boldsymbol{\mu}_j(\boldsymbol{x})$.

Definition 5.2 (Nash-Gleichgewicht [BO99, Definition 4.1])
Ein N-Tupel $\{\boldsymbol{\mu}_1^*, \ldots, \boldsymbol{\mu}_N^*\}$ stellt ein Nash-Gleichgewicht eines N-Spieler-Nicht-Nullsummen-Differenzialspiels mit unendlichem Optimierungshorizont dar, wenn $\forall i \in \mathcal{N}, \mathcal{N} := \{1, 2, \ldots, N\}$,

$$J_i(\boldsymbol{x}_0, \boldsymbol{\mu}_1^*, \ldots, \boldsymbol{\mu}_i^*, \ldots, \boldsymbol{\mu}_N^*) \leq J_i(\boldsymbol{x}_0, \boldsymbol{\mu}_1^*, \ldots, \boldsymbol{\mu}_i, \ldots, \boldsymbol{\mu}_N^*) \tag{5.3}$$

gilt.

Da in diesem Kapitel der Fokus auf der Analyse und Erfüllung der PE-Bedingung liegt, wird im Folgenden der Regulationsfall behandelt. Die Erweiterbarkeit des Differenzialspiels nach Definition 5.1 auf den Solltrajektorienfolgeregelungsfall wird jedoch in der nachfolgenden Bemerkung skizziert.

Bemerkung 5.1
ADP-kompatible Solltrajektorien nach Definition 4.1, wie beispielsweise die in Abschnitt 4.2.1 vorgestellte zeitkontinuierliche Solltrajektoriendarstellung, können in diesem Differenzialspiel nach Definition 5.1 prinzipiell ebenfalls verwendet werden. Hierzu wird anstelle des Systemzustands $\boldsymbol{x}(t)$ der erweiterte Systemzustand $\tilde{\boldsymbol{x}}(t) := \begin{bmatrix} \boldsymbol{x}^\intercal(t) & \boldsymbol{\zeta}^\intercal(t) \end{bmatrix}^\intercal$ mit $\dot{\boldsymbol{\zeta}}(t) = \boldsymbol{D}\boldsymbol{\zeta}(t)$ (vgl. (4.19)*) bzw. $\dot{\boldsymbol{\zeta}}(t) = \boldsymbol{D}'\boldsymbol{\zeta}(t)$ (vgl.* (4.48)*) betrachtet. Neben einer geeigneten Anregung von $\boldsymbol{x}(t)$, die grundsätzlich, ähnlich wie im vorliegenden Kapitel, über die Stellgröße $\boldsymbol{u}(t)$ erfolgen kann, muss dabei zusätzlich $\boldsymbol{\zeta}(t)$ angeregt werden. Dies kann, insbesondere bei Off-Policy-Algorithmen, prinzipiell über die Wahl von $\boldsymbol{\zeta}(t)$ während des Trainingsvorgangs erfolgen (vgl. auch Abbildung 4.3, Abbildung 6.4 und* (6.33) *für die Anregung von Solltrajektorienparametern $\boldsymbol{\zeta}$).*

5.2 Policy Iteration für Nicht-Nullsummen-Differenzialspiele

Für die Lösung des Nicht-Nullsummen-Differenzialspiels nach Definition 5.1 soll im Folgenden der Policy-Iteration-Algorithmus für Differenzialspiele (vgl. [SLW17, Algorithm 1], [LLW14, Algorithm 1], [VL11, Algorithm 1]) genutzt werden. Ausgehend von den Gütefunktionalen (5.2) wird zunächst ein Maß definiert, das die akkumulierten Kosten beschreibt, die dem Regler i entstehen, wenn sich das System zum Zeitpunkt t im Zustand $\boldsymbol{x}(t)$ befindet und die Regelgesetze $\boldsymbol{\mu}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{\mu}_1^\intercal(\boldsymbol{x}) & \cdots & \boldsymbol{\mu}_N^\intercal(\boldsymbol{x}) \end{bmatrix}^\intercal$ angewandt werden. Damit die Gesamtkosten endlich sind und das gesuchte Maß existiert, ist die nachfolgende Definition erforderlich.

Definition 5.3 (Zulässige Regelgesetze[91] [VL11, Definition 1])
Die Regelgesetze $\boldsymbol{\mu}_i(\boldsymbol{x}), \forall i \in \mathcal{N}$, werden als zulässig bezüglich (5.2) *auf der Menge $\mathcal{X}$ bezeichnet ($\boldsymbol{\mu}(\boldsymbol{x}) = \begin{bmatrix}\boldsymbol{\mu}_1^\mathsf{T}(\boldsymbol{x}) & \cdots & \boldsymbol{\mu}_N^\mathsf{T}(\boldsymbol{x})\end{bmatrix}^\mathsf{T} \in \Psi(\mathcal{X})$), wenn gilt:*

1. $\boldsymbol{\mu}_i(\boldsymbol{x}) \in \mathcal{C}^0(\mathcal{X})$,
2. $\boldsymbol{\mu}_i(\boldsymbol{0}) = \boldsymbol{0}$,
3. $\boldsymbol{\mu}_i(\boldsymbol{x})$ *das System* (5.1) *auf $\mathcal{X}$ stabilisiert und*
4. *die Gütemaße J_i nach* (5.2) $\forall \boldsymbol{x}_0 \in \mathcal{X}$ *endlich sind.*

Für $\boldsymbol{\mu}(\boldsymbol{x}) \in \Psi(\mathcal{X})$ sind die Value Functions $V_i^{\boldsymbol{\mu}}$, $i \in \mathcal{N}$, dann analog zu (2.12) durch

$$V_i^{\boldsymbol{\mu}}(\boldsymbol{x}) \coloneqq V_i^{\boldsymbol{\mu}}(\boldsymbol{x}(t)) = \int_t^\infty r_i(\boldsymbol{x}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_N)\,\mathrm{d}\tau \tag{5.4}$$

definiert. Zudem gelte Annahme 5.1.

Annahme 5.1
Sei $V_i^{\boldsymbol{\mu}}(\boldsymbol{x}) \in \mathcal{C}^1(\mathcal{X})$.

Aus (5.4) ergeben sich die Lyapunov-Gleichungen

$$0 = r_i(\boldsymbol{x}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_N) + \left(\nabla_{\boldsymbol{x}} V_i^{\boldsymbol{\mu}}(\boldsymbol{x})\right)^\mathsf{T}(\boldsymbol{f} + \boldsymbol{g}\boldsymbol{\mu}) \tag{5.5}$$

(vgl. [VL11, LLW14]), bzw. für $\boldsymbol{\mu} = \boldsymbol{\mu}^*$ und $V_i^{\boldsymbol{\mu}}(\boldsymbol{x}) = V_i^{\boldsymbol{\mu}^*}(\boldsymbol{x}) \eqqcolon V_i^*(\boldsymbol{x})$ die HJB-Gleichungen

$$0 = r_i(\boldsymbol{x}, \boldsymbol{\mu}_1^*, \ldots, \boldsymbol{\mu}_N^*) + \left(\nabla_{\boldsymbol{x}} V_i^*(\boldsymbol{x})\right)^\mathsf{T}(\boldsymbol{f} + \boldsymbol{g}\boldsymbol{\mu}^*)\,. \tag{5.6}$$

Mithilfe der Definition der Hamilton-Funktion[92]

$$H_i(\boldsymbol{x}, \nabla_{\boldsymbol{x}} V_i, \boldsymbol{\mu}) = r_i(\boldsymbol{x}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_N) + \left(\nabla_{\boldsymbol{x}} V_i(\boldsymbol{x})\right)^\mathsf{T}(\boldsymbol{f} + \boldsymbol{g}\boldsymbol{\mu}) \tag{5.7}$$

lässt sich mit Algorithmus 5.1 ein Lösungsalgorithmus in Form einer Policy Iteration für das Differenzialspiel gemäß Definition 5.1 angeben. Dies folgt aus Satz 5.1 (vgl. [SLW17, Theorem 2]), wobei die Konvergenz gegen V_i^*, $\forall i \in \mathcal{N}$, gerade der Nash-Lösung entspricht (vgl. [KKD14, (3)]). Die Policy Iteration kann entweder fortgeführt werden oder durch eine Abbruchbedingung, beispielsweise bei Konvergenz von $V_i^{[l]}$, $\forall i \in \mathcal{N}$, beendet werden.

Satz 5.1 (Konvergenz der Policy Iteration)
Es seien $\boldsymbol{\mu}_i^{[l]}(\boldsymbol{x})$ und $V_i^{[l]}(\boldsymbol{x})$, $i \in \mathcal{N}$, wie in Algorithmus 5.1 gegeben und $\mathcal{X}$ kompakt. Dann konvergiert die Folge $V_i^{[l]}$ für $l \to \infty$ gegen V_i^*.

91 (engl.): *admissible policies.*
92 Gleichung (5.7) ist als Funktion mit beliebigem V_i und $\boldsymbol{\mu}$ zu verstehen, es muss also nicht notwendigerweise $V_i = V_i^{\boldsymbol{\mu}}$ gelten.

Beweis:
Siehe [SLW17, Theorem 2]. □

Algorithmus 5.1 Policy Iteration für Differenzialspiele ([SLW17, Algorithm 1], [LLW14, Algorithm 1], [VL11, Algorithm 1])

1: **Initialisiere** Iterationsindex $l \coloneqq 0$, Initialregler $\boldsymbol{\mu}^{[0]} \in \Psi(\mathcal{X})$.

Schritt 1 *(Policy Evaluation):*

2: $\forall i \in \mathcal{N}$, finde $V_i^{[l+1]}$, sodass gilt:

$$0 = r_i\left(\boldsymbol{x}, \boldsymbol{\mu}_1^{[l]}, \ldots, \boldsymbol{\mu}_N^{[l]}\right) + \left(\nabla_{\boldsymbol{x}} V_i^{[l+1]}(\boldsymbol{x})\right)^{\mathsf{T}} \left(\boldsymbol{f} + \boldsymbol{g}\boldsymbol{\mu}^{[l]}\right), \tag{5.8a}$$

$$V_i^{[l+1]}(\boldsymbol{0}) = 0. \tag{5.8b}$$

Schritt 2 *(Policy Improvement):*

3: $\forall i \in \mathcal{N}$, aktualisiere das Regelgesetz

$$\begin{aligned} \boldsymbol{\mu}_i^{[l+1]}(\boldsymbol{x}) &= \underset{\boldsymbol{\mu}_i(\boldsymbol{x}) \in \Psi(\mathcal{X})}{\arg\min}\; H_i\left(\boldsymbol{x}, \nabla_{\boldsymbol{x}} V_i^{[l+1]}(\boldsymbol{x}), \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_N\right) \\ &= -\frac{1}{2}\boldsymbol{R}_{ii}^{-1}\boldsymbol{g}_i^{\mathsf{T}}(\boldsymbol{x})\nabla_{\boldsymbol{x}} V_i^{[l+1]}(\boldsymbol{x}) \end{aligned} \tag{5.9}$$

4: und setze $l \coloneqq l + 1$. Gehe zu Schritt 1.

5.3 Funktionsapproximation und Anregungsbedingung

Das Lösen der nichtlinearen Lyapunov-Gleichungen (5.8) im Policy-Evaluation-Schritt in Algorithmus 5.1 stellt im Allgemeinen eine nicht-triviale Herausforderung dar (vgl. [VL11], [LLW14]). Die Value Function wird daher durch lineare Funktionsapproximation (siehe Abschnitt 2.1.3 sowie [BBdE10], [VL11] und [LLW14]) approximiert, d. h.

$$V_i^{[l]}(\boldsymbol{x}) = \boldsymbol{w}_i^{[l]\mathsf{T}} \boldsymbol{\phi}_i(\boldsymbol{x}) + \epsilon_i^{[l]}(\boldsymbol{x}) \tag{5.10}$$

mit den Basisfunktionen $\boldsymbol{\phi}_i : \mathbb{R}^n \to \mathbb{R}^{h_i}$, beschränkten Gewichtungsvektoren $\boldsymbol{w}_i^{[l]} \in \mathbb{R}^{h_i}$ und beschränkten Approximationsfehlern $\epsilon_i^{[l]}(\boldsymbol{x}) \in \mathcal{C}^1(\mathcal{X})$, die im Allgemeinen durch die endliche Dimension h_i der Funktionsapproximatoren $\boldsymbol{\phi}_i$ resultieren können (vgl. [VL10]). Zudem sei

$$V_i^*(\boldsymbol{x}) = \boldsymbol{w}_i^{*\mathsf{T}} \boldsymbol{\phi}_i(\boldsymbol{x}) + \epsilon_i(\boldsymbol{x}), \tag{5.11}$$

mit konstanten und beschränkten optimalen[93] Gewichtungsvektoren $\boldsymbol{w}_i^* \in \mathbb{R}^{h_i}$ und beschränkten Approximationsfehlern $\epsilon_i(\boldsymbol{x})$. Um die Notation übersichtlicher zu gestalten, wird zunächst die folgende Annahme getroffen[94].

Annahme 5.2
Seien $\epsilon_i^{[l]}(\boldsymbol{x}) = 0, \forall l, \forall i \in \mathcal{N}$, und somit auch $\epsilon_i(\boldsymbol{x}) = 0$.

Da die optimalen Gewichte $\boldsymbol{w}_i^*$ des Differenzialspiels und auch die optimalen Gewichte $\boldsymbol{w}_i^{[l+1]}$ des Policy-Evaluation-Schrittes (5.8) a priori unbekannt sind, werden die geschätzten Value Functions zu

$$\hat{V}_i^{[l]}(\boldsymbol{x}) = \hat{\boldsymbol{w}}_i^{[l]\mathsf{T}} \boldsymbol{\phi}_i(\boldsymbol{x}) \tag{5.12}$$

mit den geschätzten Gewichtungsvektoren $\hat{\boldsymbol{w}}_i^{[l]} \in \mathbb{R}^{h_i}$ definiert. Ziel der ADP-basierten Regler ist somit, die Gewichte $\hat{\boldsymbol{w}}_i^{[l]}$ anzupassen, um im Policy-Evaluation-Schritt (5.8) die Lösungen $\boldsymbol{w}_i^{[l+1]}$ und letztlich die Nash-Lösungen

$$\lim_{l \to \infty} \boldsymbol{w}_i^{[l]} = \boldsymbol{w}_i^* \tag{5.13}$$

(vgl. Satz 5.1) zu bestimmen. Solange jedoch $\hat{\boldsymbol{w}}_i^{[l]} \neq \boldsymbol{w}_i^{[l]}$ gilt, ist im Allgemeinen $\hat{V}_i^{[l]} \neq V_i^{[l]}$. Daher wird im Folgenden ein Gradientenabstiegsverfahren zur Adaption von $\hat{\boldsymbol{w}}_i^{[l]}$ im Policy-Evaluation-Schritt analysiert. Nachfolgend bezeichnet $\hat{\boldsymbol{\mu}}^{[l]}(\boldsymbol{x}) = \begin{bmatrix} \hat{\boldsymbol{\mu}}_1^{[l]\mathsf{T}}(\boldsymbol{x}) & \cdots & \hat{\boldsymbol{\mu}}_N^{[l]\mathsf{T}}(\boldsymbol{x}) \end{bmatrix}^\mathsf{T}$ die geschätzten Regelgesetze in der l-ten Iteration der Policy Iteration.

Die Hamilton-Funktion (5.7) kann als Maß zur Fehlerbeschreibung des Policy-Evaluation-Schrittes (5.8) betrachtet werden. Es gilt

$$\begin{aligned} H_i\left(\boldsymbol{x}, \nabla_{\boldsymbol{x}} \hat{V}_i^{[l+1]}, \hat{\boldsymbol{\mu}}^{[l]}\right) &= r_i\left(\boldsymbol{x}, \hat{\boldsymbol{\mu}}_1^{[l]}, \ldots, \hat{\boldsymbol{\mu}}_N^{[l]}\right) + \left(\nabla_{\boldsymbol{x}} \hat{V}_i^{[l+1]}\right)^\mathsf{T} \left(\boldsymbol{f} + \boldsymbol{g}\hat{\boldsymbol{\mu}}^{[l]}\right) \\ &= r_i\left(\boldsymbol{x}, \hat{\boldsymbol{\mu}}_1^{[l]}, \ldots, \hat{\boldsymbol{\mu}}_N^{[l]}\right) \\ &\quad + \hat{\boldsymbol{w}}_i^{[l+1]\mathsf{T}} \frac{\partial \boldsymbol{\phi}_i(\boldsymbol{x})}{\partial \boldsymbol{x}} \left(\boldsymbol{f}(\boldsymbol{x}) + \sum_{j=1}^{N} \boldsymbol{g}_j(\boldsymbol{x}) \hat{\boldsymbol{\mu}}_j^{[l]}\right) \\ &=: r_i + \hat{\boldsymbol{w}}_i^{[l+1]\mathsf{T}} \boldsymbol{\sigma}_i, \end{aligned} \tag{5.14}$$

[93] Als optimale Gewichte werden im Rahmen dieses Kapitels Gewichte bezeichnet, die eine Lösung im Sinne des hier gewählten Lösungskonzepts (Nash-Gleichgewicht nach Definition 5.2) für das durch (5.1) und (5.2) beschriebene Differenzialspiel darstellen.

[94] An entsprechenden Stellen werden Hinweise auf den Einfluss von $\epsilon_i^{[l]}(\boldsymbol{x}) \neq 0$ gegeben (vgl. beispielsweise Bemerkung 5.2). Zudem sei für eine Diskussion des Einflusses nicht-exakter Funktionsapproximation auf [AKL05], [VL11] und [LLW14] verwiesen. So wird beispielsweise in [VL11, Proposition 2] gezeigt, dass für $\boldsymbol{\mu}(\boldsymbol{x}) \in \Psi(\mathcal{X})$ bei geeigneter Wahl der Basisfunktionen $\boldsymbol{\phi}_i$ für $h_i \to \infty$ der Einfluss des Approximationsfehlers $\epsilon_i^{[l]}(\boldsymbol{x})$ verschwindet.

wobei die Kurzschreibweise $\boldsymbol{\sigma}_i = \left.\frac{\mathrm{d}\boldsymbol{\phi}_i(\boldsymbol{x}(t))}{\mathrm{d}t}\right|_{\boldsymbol{u}=\hat{\boldsymbol{\mu}}^{[l]}}$ eingeführt wurde. Die Minimierung des Quadrats der Hamilton-Funktion $\frac{1}{2}\left(H_i\left(\boldsymbol{x}, \nabla_{\boldsymbol{x}}\hat{V}_i^{[l+1]}, \hat{\boldsymbol{\mu}}^{[l]}\right)\right)^2$ bezüglich $\hat{\boldsymbol{w}}_i^{[l+1]}$ mittels Gradientenabstieg liefert die Adaptionsgesetze

$$\begin{aligned}\frac{\mathrm{d}\hat{\boldsymbol{w}}_i^{[l+1]}}{\mathrm{d}t} &= -\frac{\eta_i}{2}\left(\frac{\partial\left(H_i\left(\boldsymbol{x}, \nabla_{\boldsymbol{x}}\hat{V}_i^{[l+1]}, \hat{\boldsymbol{\mu}}^{[l]}\right)\right)^2}{\partial\hat{\boldsymbol{w}}_i^{[l+1]}}\right)^{\mathsf{T}} \\ &= -\eta_i H_i\left(\boldsymbol{x}, \nabla_{\boldsymbol{x}}\hat{V}_i^{[l+1]}, \hat{\boldsymbol{\mu}}^{[l]}\right)\boldsymbol{\sigma}_i \\ &= -\eta_i\left(r_i + \hat{\boldsymbol{w}}_i^{[l+1]\mathsf{T}}\boldsymbol{\sigma}_i\right)\boldsymbol{\sigma}_i\end{aligned} \tag{5.15}$$

mit den Lernraten $\eta_i \in \mathbb{R}_{>0}$.

Mit $V_i^{[l+1]} = V_i^{\hat{\boldsymbol{\mu}}^{[l]}}(\boldsymbol{x})$, d. h., wenn $V_i^{[l+1]}$ die Lösung des Policy-Evaluation-Schritts zum aktuellen Regelgesetz $\hat{\boldsymbol{\mu}}^{[l]}$ darstellt, gilt nach (5.5) und (5.10)

$$\begin{aligned}0 &= r_i\left(\boldsymbol{x}, \hat{\boldsymbol{\mu}}_1^{[l]}, \ldots, \hat{\boldsymbol{\mu}}_N^{[l]}\right) + \left(\nabla_{\boldsymbol{x}} V_i^{\hat{\boldsymbol{\mu}}^{[l]}}(\boldsymbol{x})\right)^{\mathsf{T}}\left(\boldsymbol{f} + \boldsymbol{g}\hat{\boldsymbol{\mu}}^{[l]}\right) \\ &= r_i + \left(\nabla_{\boldsymbol{x}}\left(\boldsymbol{w}_i^{[l+1]\mathsf{T}}\boldsymbol{\phi}_i(\boldsymbol{x}) + \epsilon_i^{[l+1]}(\boldsymbol{x})\right)\right)^{\mathsf{T}}\left(\boldsymbol{f} + \boldsymbol{g}\hat{\boldsymbol{\mu}}^{[l]}\right) \\ &= r_i + \boldsymbol{\sigma}_i^{\mathsf{T}}\boldsymbol{w}_i^{[l+1]} + \left(\nabla_{\boldsymbol{x}}\epsilon_i^{[l+1]}(\boldsymbol{x})\right)^{\mathsf{T}}\left(\boldsymbol{f} + \boldsymbol{g}\hat{\boldsymbol{\mu}}^{[l]}\right) \\ &=: r_i + \boldsymbol{\sigma}_i^{\mathsf{T}}\boldsymbol{w}_i^{[l+1]} + \epsilon_{\mathrm{H},i}^{[l+1]},\end{aligned} \tag{5.16}$$

wobei der Fehler $\epsilon_{\mathrm{H},i}^{[l+1]} := \left(\nabla_{\boldsymbol{x}}\epsilon_i^{[l+1]}(\boldsymbol{x})\right)^{\mathsf{T}}\left(\boldsymbol{f} + \boldsymbol{g}\hat{\boldsymbol{\mu}}^{[l]}\right)$ verschwindet, wenn eine exakte Approximation möglich ist, also $\epsilon_i^{[l+1]}(\boldsymbol{x}) = 0$ nach Annahme 5.2 gilt.

Aus der Definition des Gewichtsfehlers $\widetilde{\boldsymbol{w}}_i^{[l+1]} := \boldsymbol{w}_i^{[l+1]} - \hat{\boldsymbol{w}}_i^{[l+1]}$ ergibt sich die Fehlerdynamik bei Verwendung des Gradientenabstiegs (5.15) zu

$$\begin{aligned}\frac{\mathrm{d}\widetilde{\boldsymbol{w}}_i^{[l+1]}}{\mathrm{d}t} &= -\frac{\mathrm{d}\hat{\boldsymbol{w}}_i^{[l+1]}}{\mathrm{d}t} = \eta_i\left(r_i + \hat{\boldsymbol{w}}_i^{[l+1]\mathsf{T}}\boldsymbol{\sigma}_i\right)\boldsymbol{\sigma}_i \\ &= \eta_i\boldsymbol{\sigma}_i\left(r_i + \boldsymbol{\sigma}_i^{\mathsf{T}}\left(\boldsymbol{w}_i^{[l+1]} - \widetilde{\boldsymbol{w}}_i^{[l+1]}\right)\right) \\ &= -\eta_i\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^{\mathsf{T}}\widetilde{\boldsymbol{w}}_i^{[l+1]} + \eta_i\boldsymbol{\sigma}_i\left(r_i + \boldsymbol{\sigma}_i^{\mathsf{T}}\boldsymbol{w}_i^{[l+1]}\right) \\ &= -\eta_i\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^{\mathsf{T}}\widetilde{\boldsymbol{w}}_i^{[l+1]} - \eta_i\boldsymbol{\sigma}_i\epsilon_{\mathrm{H},i}^{[l+1]},\end{aligned} \tag{5.17}$$

wobei der letzte Schritt unter Verwendung von (5.16) resultiert.

Um die Konvergenzeigenschaften der zentralen Fehlerdynamik (5.17) zu analysieren, wird zunächst die folgende Definition, welche die Anregung eines Signals mathematisch formalisiert, benötigt.

Definition 5.4 (PE-Funktionen nach [NA05, Definition 6.2])
Eine stückweise stetig differenzierbare, beschränkte Funktion (vgl. [NA05, Definition 6.1]) $\boldsymbol{\sigma} : \mathbb{R}_{\geq 0} \to \mathbb{R}^h$ *wird als persistently excited (PE) für alle* $t \geq t_0$[95] *bezeichnet, wenn es positive Konstanten* $\alpha, \alpha_{\mathrm{I}}, T \in \mathbb{R}_{>0}$ *gibt, sodass die äquivalenten Beziehungen*[96]

$$\int_t^{t+T} \boldsymbol{\sigma}(\tau)\boldsymbol{\sigma}^{\mathsf{T}}(\tau)\,\mathrm{d}\tau \succeq \alpha \boldsymbol{I} \tag{5.18}$$

und

$$\frac{1}{T}\int_t^{t+T} \left|(\boldsymbol{\sigma}(\tau))^{\mathsf{T}}\boldsymbol{e}\right|\mathrm{d}\tau \geq \alpha_{\mathrm{I}} \tag{5.19}$$

für alle $t \geq t_0$ *und jeden Einheitsvektor* $\boldsymbol{e} \in \mathbb{R}^h$ *gelten. Die Konstanten* α *bzw.* α_{I} *werden als Grad der PE-Bedingung bezeichnet.*

Basierend auf den Ergebnissen aus [NA05], [VL10] und [MN77] lassen sich damit die folgenden Konvergenzeigenschaften formulieren.

Lemma 5.1 (Konvergenz der Critic-Gewichte)
Seien $\left\{\hat{\boldsymbol{\mu}}_1^{[l]}, \ldots, \hat{\boldsymbol{\mu}}_N^{[l]}\right\}$ *zeitinvariante, nach Definition 5.3 zulässige Regelgesetze und Annahme 5.2 erfüllt. Die Critic-Gewichte* $\hat{\boldsymbol{w}}_i^{[l+1]}$ *zur Approximation der Lösung* $V_i^{[l+1]}$ *der Lyapunov-Gleichungen* (5.8) *(Policy-Evaluation-Schritt) gemäß* (5.12) *werden durch den Gradientenabstieg* (5.15) *angepasst. Dann folgt, dass die Gewichtsfehler*

$$\widetilde{\boldsymbol{w}}_i^{[l+1]} = \boldsymbol{w}_i^{[l+1]} - \hat{\boldsymbol{w}}_i^{[l+1]} \tag{5.20}$$

genau dann exponentiell gegen null konvergieren, wenn $\boldsymbol{\sigma}_i$ *PE nach Definition 5.4 ist. Dabei gilt* $\forall k \in \mathbb{N}_{\geq 0}$

$$\left\|\widetilde{\boldsymbol{w}}_i^{[l+1]}(kT_i)\right\|_2 \leq \exp\left(\frac{k}{2}\ln\rho_i\right)\left\|\widetilde{\boldsymbol{w}}_i^{[l+1]}(0)\right\|_2 \tag{5.21}$$

mit

$$\rho_i \coloneqq 1 - \frac{2T_i\eta_i\alpha_{\mathrm{I},i}^2}{\left(1 + T_i\eta_i\sigma_{i,\max}^2\right)^2}, \tag{5.22}$$

wobei $\sigma_{i,\max} \geq \|\boldsymbol{\sigma}_i\|_2$.

[95] Der Zeitpunkt t_0, ab dem die Funktion $\boldsymbol{\sigma}$ PE ist, ergibt sich üblicherweise aus dem Kontext. Im Folgenden sei $t_0 = 0$, sofern nicht anders definiert.
[96] Die Äquivalenz ergibt sich direkt aus [NA05, Theorem 2.16].

Beweis:
Unter Annahme 5.2 ist im Fall $\eta_i = 1$ nach [MN77, Theorem 1] (5.19) notwendig und hinreichend für die gleichmäßige asymptotische Stabilität der Ruhelage von (5.17). Nun ist laut (5.18) $\boldsymbol{\sigma}_i$ mit α_i und T_i genau dann PE, wenn $\bar{\boldsymbol{\sigma}}_i = \sqrt{\eta_i}\boldsymbol{\sigma}_i$ PE mit $\bar{\alpha}_i = \eta_i \alpha_i$ und $T_i, \forall \eta_i > 0$, ist. Zudem ist $\bar{\boldsymbol{\sigma}}_i$ genau dann PE, wenn

$$\frac{\mathrm{d}\widetilde{\boldsymbol{w}}_i^{[l+1]}}{\mathrm{d}t} = -\eta_i \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^\mathsf{T} \widetilde{\boldsymbol{w}}_i^{[l+1]} \tag{5.23}$$

gleichmäßig asymptotisch stabil ist (vgl. [MN77, Theorem 1]). Aufgrund der Linearität von (5.23) ist gleichmäßige asymptotische Stabilität identisch zu exponentieller Stabilität, weshalb $\boldsymbol{\sigma}_i$ genau dann PE ist, falls (5.23) exponentiell stabil ist. Die Abschätzung der Konvergenzgeschwindigkeit ρ_i in (5.22), welche die exponentielle Abnahme der Gewichtsfehler nach (5.21) charakterisiert, wird analog zu [NA05, Theorem 2.16] bestimmt (vgl. [VL10, Technical Lemma 2]). □

Bemerkung 5.2
Im Fall einer nicht exakten Approximierbarkeit der gesuchten Value Function $V_i^{[l+1]}$ durch $\boldsymbol{w}_i^{[l+1]}$, d. h., falls Annahme 5.2 nicht erfüllt ist, sondern beschränkte Approximationsfehler $\epsilon_i^{[l+1]} \leq \bar{\epsilon}_i^{[l+1]}$ bestehen bleiben, resultiert exponentielle Konvergenz von $\widetilde{\boldsymbol{w}}_i^{[l+1]}$ gegen die Nachbarschaft von null [VL10, Technical Lemma 2].

Die Policy Evaluation (5.8) kann somit durch die Adaption gemäß (5.15) bis zur Konvergenz von $\hat{\boldsymbol{w}}_i^{[l+1]}, \forall i \in \mathcal{N}$, erfolgen. Aus den approximierten Value Functions $\hat{V}_i^{[l+1]}$ nach Konvergenz des Gradientenabstiegsverfahrens gemäß (5.15) folgen nach (5.9) im Policy-Improvement-Schritt die approximierten Regelgesetze

$$\hat{\boldsymbol{\mu}}_i^{[l+1]}(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{R}_{ii}^{-1}\boldsymbol{g}_i^\mathsf{T}(\boldsymbol{x})\left(\frac{\partial \boldsymbol{\phi}_i(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^\mathsf{T} \hat{\boldsymbol{w}}_i^{[l+1]}. \tag{5.24}$$

Aufgrund des analytischen Zusammenhangs der approximierten Regelgesetze zu den Critic-Gewichten $\boldsymbol{w}_i$ sind nach Liu et al. [LLW14] somit keine gesonderten Funktionsapproximatoren zur Beschreibung der Regelgesetze (sogenannte *Actors*, vgl. Abschnitt 2.1.3) notwendig.

Damit stellt die Policy Iteration nach Algorithmus 5.1 ein Lösungsverfahren gemäß Definition 5.1 dar[97]. Da Lemma 5.1 belegt, dass die Erfüllung der PE-Bedingung nach Definition 5.4 notwendig und hinreichend für die exponentielle Konvergenz der durch $\widetilde{\boldsymbol{w}}_i^{[l+1]} = \boldsymbol{w}_i^{[l+1]} - \hat{\boldsymbol{w}}_i^{[l+1]}$, $i \in \mathcal{N}$, gegebenen Critic-Gewichtsfehler im Policy-Evaluation-Schritt unter Verwendung eines Gradientenabstiegs nach (5.15) ist, untersucht der Rest dieses Kapitels Bedingungen zur Erfüllung der PE-Eigenschaft von $\boldsymbol{\sigma}_i$.

[97] Im Fall einer nicht exakten Approximation der Value Function folgt neben der Konvergenz der Policy Evaluation gegen die Nachbarschaft von $\boldsymbol{w}_i^{[l+1]}$ (vgl. Bemerkung 5.2) auch Konvergenz der Policy Iteration gegen die Nachbarschaft der optimalen Lösung (vgl. [AKL05, Theorem 4], [VL11]).

5.4 Hinreichende Bedingungen zur Erfüllung der PE-Eigenschaft in ADP-basierten Differenzialspielen

In diesem Abschnitt werden für den im ADP-Kontext häufig verwendeten Fall polynomieller Funktionsapproximatoren Bedingungen an den Systemzustand $\boldsymbol{x}(t)$ hergeleitet, die garantieren, dass $\boldsymbol{\sigma}_i$ die PE-Eigenschaft erfüllt. Diese neuen hinreichenden Bedingungen können in ADP-basierten Problemen genutzt werden und bilden die theoretische Basis, um in Abschnitt 5.6 beispielhafte Anregungssignale zu entwerfen. Formal seien die Basisfunktionen $\boldsymbol{\phi}_i(\boldsymbol{x})$ wie folgt beschrieben.

Annahme 5.3 (Basisfunktionen $\boldsymbol{\phi}_i(\boldsymbol{x})$)
Die Elemente der Basisfunktionen $\boldsymbol{\phi}_i(\boldsymbol{x})$, $i \in \mathcal{N}$, seien $\forall \bar{h} \in \{1, \dots, h\}$ gemäß

$$\begin{aligned} \phi_{i,\bar{h}}(\boldsymbol{x}) &= \bar{f}_{i,\bar{h}} \prod_{j=1}^{n} x_j^{f_{i,j,\bar{h}}}, \\ \bar{f}_{i,\bar{h}} &\in \mathbb{R}_{\neq 0}, \\ f_{i,j,\bar{h}} &\in \mathbb{N}_{\geq 0} \end{aligned} \tag{5.25}$$

gewählt, sodass $\exists j \in \{1, \dots, n\} : f_{i,j,\bar{h}} \neq 0$ und $\bar{f}_{i,\bar{h}} < \infty$.

Um das Signal $\boldsymbol{x}$ formal mit der PE-Eigenschaft der transformierten Signale $\boldsymbol{\sigma}_i = \dot{\boldsymbol{\phi}}_i(\boldsymbol{x})$ in Zusammenhang zu bringen, wird der Begriff *sufficiently rich* (SR)[98] definiert.

Definition 5.5 (SR-Signal)
Sei $\mathcal{S}\{\cdot\}$ die Transformation eines Signals, die aus der Zusammenschaltung von Integratoren, Differenziatoren und nichtlinearen, Lipschitz-stetigen Funktionen bestehen kann. Das Signal $\boldsymbol{x}(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ wird als sufficiently rich (SR) bezüglich $\mathcal{S}$ bezeichnet, wenn $\mathcal{S}\{\boldsymbol{x}(t)\}$ PE ist.

Somit ist $\boldsymbol{\sigma}_i$ PE, wenn $\boldsymbol{x}(t)$ SR bezüglich $\dot{\boldsymbol{\phi}}_i(\boldsymbol{x})$ ist. Daraus resultiert die folgende Problemstellung für dieses Unterkapitel.

Problem 5.1
Sei $\boldsymbol{\phi}_i(\boldsymbol{x})$ $\forall i \in \mathcal{N}$ wie in Annahme 5.3. Gesucht ist $\boldsymbol{x}(t)$, sodass $\forall i \in \mathcal{N}$ das durch $\boldsymbol{\sigma}_i = \dot{\boldsymbol{\phi}}_i(\boldsymbol{x}(t))$ gegebene Signal die PE-Eigenschaft nach Definition 5.4 erfüllt, d. h. $\boldsymbol{x}(t)$ SR bezüglich $\dot{\boldsymbol{\phi}}_i(\boldsymbol{x})$ ist.

[98] Boyd und Sastry [BS86] definieren den Begriff der SR-Signale für lineare Systeme (im Zusammenhang des *Model Reference Adaptive Control*) mit skalarem Eingang. Die in der vorliegenden Arbeit vorgestellte Definition kann daher als Verallgemeinerung auf den mehrdimensionalen und nichtlinearen Fall verstanden werden.

Da das zur Adaption verwendete Signal $\boldsymbol{\sigma}_i = \dot{\boldsymbol{\phi}}_i(\boldsymbol{x})$ aus nichtlinearen Basisfunktionen $\boldsymbol{\phi}_i(\boldsymbol{x})$ resultiert[99], kann diese Transformation von $\boldsymbol{x}(t)$ nicht mithilfe klassischer Eigenschaften linearer Transformationen aus der PE-Literatur [NA05], [NA87] analysiert werden. So kann, abhängig von $\boldsymbol{\phi}_i(\boldsymbol{x})$, das Signal $\boldsymbol{\sigma}_i$ PE sein, obwohl $\boldsymbol{x}$ nicht PE ist, oder umgekehrt [LK98], [LK99]. Dies ist in Beispiel 5.1 illustriert.

Beispiel 5.1

1. ***$\boldsymbol{x}$ ist PE $\not\Rightarrow$ $\boldsymbol{\sigma}_i$ ist PE***
   Mit der PE-Funktion

$$\boldsymbol{x}(t) = \begin{bmatrix} \frac{1}{2}\cos(2\omega t) \\ \cos(\omega t) \end{bmatrix}, \ \omega \neq 0, \ \textit{und} \ \boldsymbol{\phi}_i(\boldsymbol{x}) = \begin{bmatrix} x_1 \\ x_2^2 \end{bmatrix} \tag{5.26}$$

   folgt, dass

$$\boldsymbol{\sigma}_i = \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_2^2 \end{bmatrix} = \begin{bmatrix} -\omega\sin(2\omega t) \\ -\omega\sin(2\omega t) \end{bmatrix} \tag{5.27}$$

   keine PE-Funktion ist.

2. ***$\boldsymbol{x}$ ist PE $\not\Leftarrow$ $\boldsymbol{\sigma}_i$ ist PE***
   Zwar stellt

$$\boldsymbol{x}(t) = \begin{bmatrix} \sin(\omega t) \\ 0 \end{bmatrix}, \ \omega \neq 0, \tag{5.28}$$

   keine PE-Funktion dar, aus

$$\boldsymbol{\phi}_i(\boldsymbol{x}) = \begin{bmatrix} x_1 \\ x_1^2 \end{bmatrix} \ \textit{und} \ \boldsymbol{\sigma}_i = \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_1^2 \end{bmatrix} = \begin{bmatrix} \omega\cos(\omega t) \\ \omega\sin(2\omega t) \end{bmatrix} \tag{5.29}$$

   folgt jedoch, dass $\boldsymbol{\sigma}_i$ eine PE-Funktion ist.

Somit ist die Erfüllung der PE-Eigenschaft von $\boldsymbol{x}$ weder notwendig noch hinreichend dafür, dass $\boldsymbol{\sigma}_i$ PE ist. Beispiel 5.1 veranschaulicht, dass durch polynomielle Basisfunktionen $\boldsymbol{\phi}_i(\boldsymbol{x})$ aufgrund deren nichtlinearen Charakters im Vergleich zu $\boldsymbol{x}$ sowohl Frequenzen ausgelöscht, als auch zusätzliche Frequenzen erzeugt werden können. Motiviert durch diese Erkenntnis ist die zentrale Idee im Folgenden, diese Frequenzen zu analysieren. Um Trajektorien $\boldsymbol{x}(t)$ zu finden, die eine Lösung von Problem 5.1 darstellen, wird zunächst die folgende allgemeine Struktur einer Anregungstrajektorie $\boldsymbol{x}_{\mathrm{PE}}(t)$ angesetzt.

99 Selbst für den Spezialfall eines linear-quadratischen Differenzialspiels mit linearer Systemdynamik und quadratischen Gütefunktionalen ist $\boldsymbol{\phi}_i(\boldsymbol{x})$ nichtlinear.

Annahme 5.4 (Anregungstrajektorie $\boldsymbol{x}_{\mathrm{PE}}(t)$)
Die Anregungstrajektorie $\boldsymbol{x}_{\mathrm{PE}}(t) = \left[x_{\mathrm{PE},1}(t), \dots, x_{\mathrm{PE},n}(t)\right]^{\mathsf{T}}$ des Systemzustands $\boldsymbol{x}(t)$ wird $\forall o \in \{1, \dots, n\}$ zu

$$x_{\mathrm{PE},o} = \sum_{j=1}^{m} g_{j,o} \sin(\omega_j t) + \sum_{j=1}^{m} \bar{g}_{j,o} \cos(\omega_j t), \tag{5.30}$$

$g_{j,o}$, $\bar{g}_{j,o} \in \mathbb{R}$, gewählt, sodass

1. *$\forall o \in \{1, \dots, n\}\ \exists j \in \{1, \dots, m\} : g_{j,o} \neq 0$ oder $\bar{g}_{j,o} \neq 0$,*
2. *$|g_{j,o}| < \infty$ und $|\bar{g}_{j,o}| < \infty$ und*
3. *$1 \leq m < \infty$.*

Hierbei sind $\omega_1, \dots, \omega_m$ Frequenzvariablen, die den Frequenzvektor $\boldsymbol{\omega} = \begin{bmatrix}\omega_1 & \dots & \omega_m\end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^m$ bilden.

In den nachfolgenden Schritten wird der Spielerindex i o. B. d. A. vernachlässigt, um die Notation übersichtlicher zu gestalten. Zum Ende des Abschnitts werden schließlich wieder alle N Spieler berücksichtigt. Der Frequenzvektor $\boldsymbol{\omega}$ soll nun so gewählt werden, dass $\boldsymbol{x}_{\mathrm{PE}}(t)$ SR bezüglich $\dot{\boldsymbol{\phi}}(\boldsymbol{x})$ ist. Unter Annahme 5.3 und Annahme 5.4 resultiert das folgende Lemma, das die in $\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{PE}})$ auftretenden Frequenzen beschreibt.

Lemma 5.2
Werden $\boldsymbol{\phi}(\boldsymbol{x})$ und $\boldsymbol{x}_{\mathrm{PE}}(t)$ gemäß Annahme 5.3 und Annahme 5.4 gewählt, so ergeben sich die Elemente von $\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{PE}})$ zu

$$\phi_{\bar{h}}(\boldsymbol{x}_{\mathrm{PE}}) = \sum_{l=1}^{L_{\bar{h}}} a_{l,\bar{h}} \sin\left(\underline{\omega}_{l,\bar{h}} t\right) + \sum_{k=1}^{K_{\bar{h}}} c_{k,\bar{h}} \cos\left(\bar{\underline{\omega}}_{k,\bar{h}} t\right) + e_{\bar{h}}, \tag{5.31}$$

$\forall \bar{h} \in \{1, \dots, h\}$, mit $a_{l,\bar{h}}, c_{k,\bar{h}} \in \mathbb{R}_{\neq 0}$, $e_{\bar{h}} \in \mathbb{R}$ und

$$\underline{\omega}_{l,\bar{h}} = \sum_{j=1}^{m} b_{j,l,\bar{h}} \omega_j, \tag{5.32}$$

$$\bar{\underline{\omega}}_{k,\bar{h}} = \sum_{j=1}^{m} d_{j,k,\bar{h}} \omega_j, \tag{5.33}$$

wobei $b_{j,l,\bar{h}}, d_{j,k,\bar{h}} \in \mathbb{R}$. Zudem gilt für die oberen Summengrenzen $L_{\bar{h}}, K_{\bar{h}} \in \mathbb{N}_{\geq 0}$

$$L_{\bar{h}} + K_{\bar{h}} \geq 1, \qquad \forall \bar{h} \in \{1, \dots, h\}. \tag{5.34}$$

Beweis:
Der Beweis ist in Anhang C.1 skizziert. □

Im Folgenden werden durch eine Analyse der Frequenzen $\underline{\omega}_{l,\bar{h}}$ nach (5.32) und $\underline{\bar{\omega}}_{k,\bar{h}}$ nach (5.33), die aus der nichtlinearen Transformation $\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{PE}})$ resultieren, ausschließende Bedingungen an die Frequenzen $\boldsymbol{\omega}$ hergeleitet, sodass $\dot{\boldsymbol{\phi}}(\boldsymbol{x}_{\mathrm{PE}})$ PE ist. Hierbei wird das nachfolgende notwendige und hinreichende Kriterium für mehrdimensionale PE-Signale genutzt.

Lemma 5.3
Sei $\boldsymbol{\sigma}(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^h$. Das Signal $\boldsymbol{\sigma}(t)$ ist genau dann PE, wenn $\boldsymbol{\alpha}^\intercal \boldsymbol{\sigma}(t)$ für jeden konstanten Vektor $\boldsymbol{\alpha} \in \mathbb{R}^h$ mit $\boldsymbol{\alpha} \neq \mathbf{0}$ mindestens eine Spektrallinie im Frequenzspektrum aufweist.

Beweis:
Das Signal $\boldsymbol{\sigma}(t)$ ist genau dann PE, wenn $\boldsymbol{\alpha}^\intercal \boldsymbol{\sigma}(t)$ für jeden konstanten Vektor $\boldsymbol{\alpha} \in \mathbb{R}^h$, $\boldsymbol{\alpha} \neq \mathbf{0}$, PE ist [NA05, Lemma 6.2]. Des Weiteren ist ein skalares Signal $\sigma_{\mathrm{s}}(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}$ genau dann PE, wenn es mindestens eine Spektrallinie aufweist [NA05, Sublemma 6.1]. Mit $\sigma_{\mathrm{s}}(t) := \boldsymbol{\alpha}^\intercal \boldsymbol{\sigma}(t)$ folgt somit die Aussage von Lemma 5.3. □

Basierend auf Lemma 5.3 wird nun eine Menge Ω definiert, sodass für jeden Vektor $\boldsymbol{\omega} \in \Omega$ die Erfüllung der PE-Bedingung von $\boldsymbol{\sigma} = \dot{\boldsymbol{\phi}}(\boldsymbol{x}_{\mathrm{PE}})$ garantiert wird. Die nachfolgende Proposition charakterisiert diese Menge Ω.

Proposition 5.1
Seien $\boldsymbol{\phi}(\boldsymbol{x})$ und $\boldsymbol{x}_{\mathrm{PE}}$ in Übereinstimmung zu Annahme 5.3 und Annahme 5.4 gewählt. Dann ist $\boldsymbol{x}_{\mathrm{PE}}$ SR bezüglich $\dot{\boldsymbol{\phi}}(\boldsymbol{x}_{\mathrm{PE}})$ und somit $\boldsymbol{\sigma} = \dot{\boldsymbol{\phi}}(\boldsymbol{x}_{\mathrm{PE}})$ PE $\forall \boldsymbol{\omega} \in \Omega$, wenn eine nichtleere Menge $\Omega \subseteq \mathbb{R}^m$ existiert, sodass $\forall \boldsymbol{\omega} \in \Omega$ und jeden konstanten Vektor $\boldsymbol{\alpha} \in \mathbb{R}^h$ mit $\boldsymbol{\alpha} \neq \mathbf{0}$

$$\boldsymbol{\alpha}^\intercal \boldsymbol{\sigma} = \boldsymbol{\alpha}^\intercal \dot{\boldsymbol{\phi}}(\boldsymbol{x}_{\mathrm{PE}}) = \sum_{l=1}^{L^{\langle\alpha\rangle}} a_l^{\langle\alpha\rangle} \cos\Big(\underline{\omega}_l^{\langle\alpha\rangle} t\Big) + \sum_{k=1}^{K^{\langle\alpha\rangle}} c_k^{\langle\alpha\rangle} \sin\Big(\underline{\bar{\omega}}_k^{\langle\alpha\rangle} t\Big) \tag{5.35}$$

mit $L^{\langle\alpha\rangle} + K^{\langle\alpha\rangle} \geq 1$ und

$$\begin{aligned} a_l^{\langle\alpha\rangle} &\neq 0, & \underline{\omega}_l^{\langle\alpha\rangle} &\neq 0, & \underline{\omega}_{l_1}^{\langle\alpha\rangle} &\neq \underline{\omega}_{l_2}^{\langle\alpha\rangle}, & \underline{\omega}_{l_1}^{\langle\alpha\rangle} &\neq -\underline{\omega}_{l_2}^{\langle\alpha\rangle}, \\ c_k^{\langle\alpha\rangle} &\neq 0, & \underline{\bar{\omega}}_k^{\langle\alpha\rangle} &\neq 0, & \underline{\bar{\omega}}_{k_1}^{\langle\alpha\rangle} &\neq \underline{\bar{\omega}}_{k_2}^{\langle\alpha\rangle}, & \underline{\bar{\omega}}_{k_1}^{\langle\alpha\rangle} &\neq -\underline{\bar{\omega}}_{k_2}^{\langle\alpha\rangle}, \end{aligned}$$

$\forall l_{1/2} \in \{1, \ldots, L^{\langle\alpha\rangle}\}, l_1 \neq l_2$ und $\forall k_{1/2} \in \{1, \ldots, K^{\langle\alpha\rangle}\}$, $k_1 \neq k_2$ gilt. Der Index $\langle\alpha\rangle$ kennzeichnet hierbei eine Abhängigkeit von $\boldsymbol{\alpha}$.

Beweis:
Das Signal auf der rechten Seite von (5.35) weist unter den in Proposition 5.1 gegebenen Bedingungen mindestens eine Spektrallinie im Frequenzspektrum auf. Dies wird, neben dem Auftreten mindestens einer Frequenz ($L^{\langle\alpha\rangle} + K^{\langle\alpha\rangle} \geq 1$), durch die lineare Unabhängigkeit von Sinusfunktionen (bzw. Kosinusfunktionen) unterschiedlicher Frequenzen sowie der linearen Unabhängigkeit einer Sinus- zur Kosinusfunktion derselben Frequenz gewährleistet. Da (5.35) für jeden konstanten Vektor $\boldsymbol{\alpha} \in \mathbb{R}^h$, $\boldsymbol{\alpha} \neq \mathbf{0}$, gilt, folgt mit Lemma 5.3, dass $\boldsymbol{\sigma} = \dot{\boldsymbol{\phi}}(\boldsymbol{x}_{\mathrm{PE}})$ PE ist. □

Um nun eine Anregungstrajektorie $\boldsymbol{x}_{\mathrm{PE}}(t)$ zu finden, die SR bezüglich $\dot{\boldsymbol{\phi}}(\boldsymbol{x}_{\mathrm{PE}})$ ist, genügt es gemäß Proposition 5.1, eine nichtleere Menge Ω zu finden, die Bedingungen an die Frequenzen $\boldsymbol{\omega}$ in $\boldsymbol{x}_{\mathrm{PE}}$ so formuliert, dass (5.35) erfüllt ist. Die analytischen Zusammenhänge (5.32) und (5.33) zwischen den Frequenzen ω_j in $\boldsymbol{x}_{\mathrm{PE}}$ und den durch die nichtlineare Transformation eingeführten Frequenzen $\underline{\omega}_{l,\bar{h}}$ und $\bar{\underline{\omega}}_{k,\bar{h}}$ dienen hierzu im Folgenden als Ausgangspunkt. Dies lässt sich motivieren, da die zeitliche Ableitung von (5.31) nur den Typ der trigonometrischen Funktionen sowie deren Koeffizienten verändert. Die in $\sigma_{\bar{h}} = \dot{\phi}_{\bar{h}}(\boldsymbol{x}_{\mathrm{PE}})$, $\forall \bar{h} \in \{1, \ldots, h\}$, auftretenden Frequenzen folgen direkt aus (5.31). Durch die zeitliche Ableitung von $\phi_{\bar{h}}(\boldsymbol{x}_{\mathrm{PE}})$ entfällt außerdem die Konstante $e_{\bar{h}}$ und muss daher nicht weiter berücksichtigt werden. Zudem sind die genauen Werte der Koeffizienten $a_{l,\bar{h}}$ und $c_{k,\bar{h}}$ in (5.31) irrelevant für die Erfüllung von (5.35). Dies gilt, da gemäß Proposition 5.1 aufgrund von $L^{\langle\alpha\rangle} + K^{\langle\alpha\rangle} \geq 1$ für mindestens eine Frequenz eine Amplitude ungleich null resultiert.

Im Folgenden wird nun eine mögliche Menge Ω an Frequenzbedingungen gemäß Proposition 5.1 berechnet. Hierzu werden in den Abschnitten 5.4.1 und 5.4.2 zunächst die Hilfsmengen $\Omega^{\langle 1\rangle}$ und $\Omega^{\langle 2\rangle}$ definiert und vereinfacht. In Abschnitt 5.4.3 folgt schließlich die Menge

$$\Omega = \Omega^{\langle 1\rangle} \cap \Omega^{\langle 2\rangle} \tag{5.36}$$

an Frequenzbedingungen sowie die theoretische Aussage, dass die durch diese Menge Ω gegebenen Bedingungen die Erfüllung von (5.35) sicherstellen.

5.4.1 Hilfsmenge $\Omega^{\langle 1\rangle}$

Die nachfolgende Definition formuliert durch die Hilfsmenge $\Omega^{\langle 1\rangle}$ Bedingungen an $\boldsymbol{\omega}$. Diese garantieren, dass sich, unabhängig von den exakten Werten der Koeffizienten und Argumente, für keinen Vektor $\boldsymbol{\omega} \in \Omega^{\langle 1\rangle}$ die trigonometrischen Funktionen innerhalb eines Elements $\phi_{\bar{h}}(\boldsymbol{x}_{\mathrm{PE}})$ und somit in $\sigma_{\bar{h}}$ kompensieren.

Definition 5.6 ($\Omega^{\langle 1 \rangle}$: Einzigartigkeit der Frequenzen innerhalb jedes Elements $\phi_{\bar{h}}(\boldsymbol{x}_{\mathrm{PE}}), \forall \bar{h} \in \{1, \ldots, h\}$)
Die Menge $\Omega^{\langle 1 \rangle}$ an Bedingungen an die Frequenzvariable $\boldsymbol{\omega}$ sei durch

$$\Omega^{\langle 1 \rangle} := \left\{ \boldsymbol{\omega} : \bigwedge_{\bar{h}=1}^{h} \left(\bigwedge_{l_1=1}^{L_{\bar{h}}-1} \bigwedge_{l_2>l_1}^{L_{\bar{h}}} \left(\underline{\omega}_{l_1,\bar{h}} \neq \underline{\omega}_{l_2,\bar{h}} \wedge \underline{\omega}_{l_1,\bar{h}} \neq -\underline{\omega}_{l_2,\bar{h}} \right) \right.\right. \\ \left.\left. \wedge \bigwedge_{k_1=1}^{K_{\bar{h}}-1} \bigwedge_{k_2>k_1}^{K_{\bar{h}}} \left(\underline{\bar{\omega}}_{k_1,\bar{h}} \neq \underline{\bar{\omega}}_{k_2,\bar{h}} \wedge \underline{\bar{\omega}}_{k_1,\bar{h}} \neq -\underline{\bar{\omega}}_{k_2,\bar{h}} \right) \right) \right\} \tag{5.37}$$

definiert.

Somit sind für jedes $\boldsymbol{\omega} \in \Omega^{\langle 1 \rangle}$ nach Definition 5.6 in jedem Element $\phi_{\bar{h}}(\boldsymbol{x}_{\mathrm{PE}})$, d. h. für jedes $\bar{h} \in \{1, \ldots, h\}$, die Beträge der Frequenzen der Sinusterme einzigartig, Gleiches gilt für die Kosinusterme. Die Menge $\Omega^{\langle 1 \rangle}$ kann wie folgt vereinfacht werden.

Lemma 5.4 (Vereinfachung der Menge $\Omega^{\langle 1 \rangle}$)
Die durch Definition 5.6 beschriebene Menge $\Omega^{\langle 1 \rangle}$ lässt sich zu

$$\Omega^{\langle 1 \rangle} = \left\{ \boldsymbol{\omega} : \bigwedge_{z=1}^{Z_1} \boldsymbol{c}_z^{\mathsf{T}} \boldsymbol{\omega} \neq 0 \right\}, \quad Z_1 \in \mathbb{N}_{\geq 0}, \tag{5.38}$$

umformulieren.

Beweis:
Ausgehend von (5.37) wird zunächst die Komplementärmenge $\bar{\Omega}^{\langle 1 \rangle}$ gebildet. Anschließendes Anwenden der De Morganschen Gesetze [BSMM13, S. 332] führt auf eine Menge, die durch die Disjunktion linearer algebraischer Gleichungen der Form $\underline{\omega}_{l_1,\bar{h}} = \underline{\omega}_{l_2,\bar{h}}$ gegeben ist. Diese Gleichungen können in Matrixschreibweise $\bar{\boldsymbol{c}}_z^{\mathsf{T}} \boldsymbol{\omega} = 0$ mit $\bar{\boldsymbol{c}}_z \in \mathbb{R}^m$ überführt werden, wobei z einen Laufindex darstellt.

Durch Skalierung der Zeilenvektoren $\bar{\boldsymbol{c}}_z$ können diese Ausdrücke jeweils auf reduzierte Zeilenstufenform gebracht werden. Die Transformation der Koeffizientenmatrix eines homogenen linearen Gleichungssystems auf reduzierte Zeilenstufenform verändert dessen Lösungsmenge nicht [KB18, S. 22]. Dies gilt auch für $\bar{\boldsymbol{c}}_z^{\mathsf{T}} \boldsymbol{\omega} = 0$ und dessen reduzierte Zeilenstufenform $\boldsymbol{c}_z^{\mathsf{T}} \boldsymbol{\omega} = 0$. Werden anschließend alle mehrfach auftretenden Bedingungen eliminiert und abermals die De Morganschen Gesetze angewandt, so resultiert (5.38). Da diese mehrfach auftretenden Bedingungen die Menge $\bar{\Omega}^{\langle 1 \rangle}$ nicht verändern, ist $\Omega^{\langle 1 \rangle}$ (5.37) identisch zu (5.38). □

5.4.2 Hilfsmenge $\Omega^{\langle 2\rangle}$

Durch (5.35) in Proposition 5.1 werden Bedingungen an die gewichtete Summe der Elemente von $\boldsymbol{\sigma} = \dot{\boldsymbol{\phi}}(\boldsymbol{x}_{\mathrm{PE}})$ formuliert. Daher genügt eine separierte Betrachtung der Elemente $\phi_{\bar{h}}(\boldsymbol{x}_{\mathrm{PE}})$, $\bar{h} \in \{1, \ldots, h\}$, nicht und die Hilfsmenge $\Omega^{\langle 2\rangle}$ analysiert im Folgenden Frequenzen verschiedener Einträge von $\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{PE}})$. Bevor die Menge $\Omega^{\langle 2\rangle}$ an Frequenzbedingungen formal definiert wird, werden zunächst die Hilfsmatrizen $\boldsymbol{P}^{\mathrm{f}}$ und $\boldsymbol{T}^{\mathrm{f}}$ gegeben. Die Spalten der Matrix $\boldsymbol{P}^{\mathrm{f}}$ beinhalten dabei Kombinationen der Frequenzen zwischen den verschiedenen Elementen $\phi_{\bar{h}}(\boldsymbol{x}_{\mathrm{PE}})$ und $\boldsymbol{T}^{\mathrm{f}}$ indiziert den jeweils zugehörigen Typ der trigonometrischen Funktion.

Definition 5.7 (Repräsentation von Frequenztupeln durch $\boldsymbol{P}^{\mathrm{f}}$ und $\boldsymbol{T}^{\mathrm{f}}$)
Seien

$$W_{\bar{h}} := \left\{ \underline{\omega}_{1,\bar{h}}, \ldots, \underline{\omega}_{L_{\bar{h}},\bar{h}}, \bar{\omega}_{1,\bar{h}}, \ldots, \bar{\omega}_{K_{\bar{h}},\bar{h}} \right\}, \tag{5.39}$$

$\bar{h} \in \{1, \ldots, h\}$, *Mengen, welche die Frequenzen nach* (5.32) *und* (5.33) *beinhalten. Dann lassen sich mit* $\zeta_{\bar{h}}^{\langle i\rangle} \in W_{\bar{h}}$, $\bar{h} \in \{1, \ldots, h\}$,

$$N_{\mathrm{P}} = \prod_{\bar{h}=1}^{h} (L_{\bar{h}} + K_{\bar{h}}) \tag{5.40}$$

unterschiedliche Tupel $\mathcal{P}_i := \left(\zeta_1^{\langle i\rangle}, \ldots, \zeta_h^{\langle i\rangle}\right)$, $i = 1, \ldots, N_{\mathrm{P}}$, *definieren. Weiterhin sei* $\boldsymbol{P}^{\mathrm{f}}$ *eine* $h \times N_{\mathrm{P}}$*-Matrix, deren* N_{P} *Spalten die Einträge der Tupel* $\mathcal{P}_1, \ldots, \mathcal{P}_{N_{\mathrm{P}}}$ *enthalten. Die Information über den jeweils zugehörigen Typ der trigonometrischen Funktion wird in der Matrix* $\boldsymbol{T}^{\mathrm{f}} \in \mathbb{R}^{h \times N_{\mathrm{P}}}$ *codiert. Hierbei wird das* $(\bar{h}, n_{\mathrm{P}})$*-te Element von* $\boldsymbol{T}^{\mathrm{f}}$ *zu null gesetzt* ($T^{\mathrm{f}}_{\bar{h},n_{\mathrm{P}}} = 0$)*, wenn das* $(\bar{h}, n_{\mathrm{P}})$*-te Element* $P^{\mathrm{f}}_{\bar{h},n_{\mathrm{P}}}$ *von* $\boldsymbol{P}^{\mathrm{f}}$ *das Argument einer Sinusfunktion darstellt. Ist es hingegen das Argument einer Kosinusfunktion, so gilt* $T^{\mathrm{f}}_{\bar{h},n_{\mathrm{P}}} = 1$.

Gemäß Definition 5.7 lassen sich folglich o. B. d. A. die Matrizen

$$\boldsymbol{P}^{\mathrm{f}} = \begin{bmatrix} \underline{\omega}_{1,1} & \cdots & \underline{\omega}_{1,1} & \cdots & \underline{\omega}_{1,1} & \cdots & \underline{\omega}_{1,1} & \cdots & \bar{\omega}_{K_1,1} & \cdots & \bar{\omega}_{K_1,1} \\ \underline{\omega}_{1,2} & \cdots & \underline{\omega}_{1,2} & \cdots & \bar{\omega}_{K_2,2} & \cdots & \bar{\omega}_{K_2,2} & \cdots & \bar{\omega}_{K_2,2} & \cdots & \bar{\omega}_{K_2,2} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \underline{\omega}_{1,h} & \cdots & \bar{\omega}_{K_h,h} & \cdots & \underline{\omega}_{1,h} & \cdots & \bar{\omega}_{K_h,h} & \cdots & \underline{\omega}_{1,h} & \cdots & \bar{\omega}_{K_h,h} \end{bmatrix} \tag{5.41}$$

und

$$\boldsymbol{T}^{\mathrm{f}} = \begin{bmatrix} 0 & \ldots & 0 & \ldots & 0 & \ldots & 0 & \ldots & 1 & \ldots & 1 \\ 0 & \ldots & 0 & \ldots & 1 & \ldots & 1 & \ldots & 1 & \ldots & 1 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \ldots & 1 & \ldots & 0 & \ldots & 1 & \ldots & 0 & \ldots & 1 \end{bmatrix} \tag{5.42}$$

definieren. Damit lässt sich nun $\Omega^{\langle 2\rangle}$ formulieren.

Definition 5.8 **($\Omega^{\langle 2 \rangle}$: Bedingungen an die Frequenzen in $\boldsymbol{P}^{\mathrm{f}}$)**
Mit $\boldsymbol{P}^{\mathrm{f}}$ (5.41) *und* $\boldsymbol{T}^{\mathrm{f}}$ (5.42) *nach Definition 5.7 sei*

$$\begin{aligned}\Omega^{\langle 2 \rangle} := \Bigg\{ \boldsymbol{\omega} : \bigvee_{n_{\mathrm{P}}=1}^{N_{\mathrm{P}}} \Bigg(& \underbrace{\bigwedge_{\bar{h}=1}^{h} P^{\mathrm{f}}_{\bar{h},n_{\mathrm{P}}} \neq 0}_{\mathrm{I}} \\ & \wedge \underbrace{\bigwedge_{\bar{h}_1=1}^{h-1} \bigwedge_{\bar{h}_2 \in \bar{H}} \left(P^{\mathrm{f}}_{\bar{h}_1,n_{\mathrm{P}}} \neq P^{\mathrm{f}}_{\bar{h}_2,n_{\mathrm{P}}} \wedge P^{\mathrm{f}}_{\bar{h}_1,n_{\mathrm{P}}} \neq -P^{\mathrm{f}}_{\bar{h}_2,n_{\mathrm{P}}} \right)}_{\mathrm{II}} \\ & \wedge \underbrace{\bigwedge_{\substack{\bar{h}_{1/2}=1 \\ \bar{h}_1 \neq \bar{h}_2}}^{h} \bigwedge_{\bar{l} \in \bar{L}} \left(P^{\mathrm{f}}_{\bar{h}_1,n_{\mathrm{P}}} \neq \underline{\omega}_{\bar{l},\bar{h}_2} \wedge P^{\mathrm{f}}_{\bar{h}_1,n_{\mathrm{P}}} \neq -\underline{\omega}_{\bar{l},\bar{h}_2} \right)}_{\mathrm{III_a}} \\ & \wedge \underbrace{\bigwedge_{\substack{\bar{h}_{1/2}=1 \\ \bar{h}_1 \neq \bar{h}_2}}^{h} \bigwedge_{\bar{k} \in \bar{K}} \left(P^{\mathrm{f}}_{\bar{h}_1,n_{\mathrm{P}}} \neq \underline{\bar{\omega}}_{\bar{k},\bar{h}_2} \wedge P^{\mathrm{f}}_{\bar{h}_1,n_{\mathrm{P}}} \neq -\underline{\bar{\omega}}_{\bar{k},\bar{h}_2} \right)}_{\mathrm{III_b}} \Bigg) \Bigg\} \end{aligned} \tag{5.43}$$

mit den Indexmengen

$$\begin{aligned} \bar{H} := \bar{H}(\bar{h}_1, n_{\mathrm{P}}) &:= \left\{ \bar{h}_2 \in \{1, \dots, h\} : \bar{h}_2 > \bar{h}_1 \wedge T^{\mathrm{f}}_{\bar{h}_1,n_{\mathrm{P}}} = T^{\mathrm{f}}_{\bar{h}_2,n_{\mathrm{P}}} \right\}, \\ \bar{L} := \bar{L}(\bar{h}_1, \bar{h}_2, n_{\mathrm{P}}) &:= \left\{ \bar{l} \in \{1, \dots, L_{\bar{h}_2}\} : \underline{\omega}_{\bar{l},\bar{h}_2} \neq P^{\mathrm{f}}_{\bar{h}_2,n_{\mathrm{P}}} \wedge T^{\mathrm{f}}_{\bar{h}_1,n_{\mathrm{P}}} = 0 \right\}, \\ \bar{K} := \bar{K}(\bar{h}_1, \bar{h}_2, n_{\mathrm{P}}) &:= \left\{ \bar{k} \in \{1, \dots, K_{\bar{h}_2}\} : \underline{\bar{\omega}}_{\bar{k},\bar{h}_2} \neq P^{\mathrm{f}}_{\bar{h}_2,n_{\mathrm{P}}} \wedge T^{\mathrm{f}}_{\bar{h}_1,n_{\mathrm{P}}} = 1 \right\}. \end{aligned} \tag{5.44}$$

Somit muss nach Definition 5.8 mindestens eine der N_{P} Frequenzkombinationen, die durch die N_{P} Spalten von $\boldsymbol{P}^{\mathrm{f}}$ (5.41) gegeben sind, die Bedingungen innerhalb der äußersten Disjunktion in (5.43) erfüllen. Die zu I zusammengefassten Bedingungen fordern, dass die durch die n_{P}-te Spalte von $\boldsymbol{P}^{\mathrm{f}}$ beschriebenen Frequenzterme nicht verschwinden. Bedingung II garantiert, dass keine zwei trigonometrischen Funktionen desselben Typs in der betrachteten Spalte von $\boldsymbol{P}^{\mathrm{f}}$ dieselbe Frequenz aufweisen. Bedingung $\mathrm{III} = \mathrm{III_a} \wedge \mathrm{III_b}$ gewährleistet schließlich, dass der Frequenzterm $P^{\mathrm{f}}_{\bar{h}_1,n_{\mathrm{P}}}$ der betrachteten Frequenzkombination, d. h. der n_{P}-ten Spalte von $\boldsymbol{P}^{\mathrm{f}}$, ungleich zu den Frequenztermen desselben trigonometrischen Typs in den anderen Elementen $\phi_{\bar{h}_2}(\boldsymbol{x}_{\mathrm{PE}})$, $\bar{h}_2 \neq \bar{h}_1$, ist. Ungleichheit impliziert hier, genau wie in Definition 5.6, auch die Ungleichheit zu Frequenztermen mit gegensätzlichem Vorzeichen.

Analog zu Lemma 5.4 kann die Menge $\Omega^{\langle 2\rangle}$ (5.43) vereinfacht werden. Hierzu wird zunächst die Komplementärmenge $\bar{\Omega}^{\langle 2\rangle}$, die aus der Anwendung der De Morganschen Gesetze resultiert, mithilfe von Algorithmus 5.2 vereinfacht. Daraus ergibt sich[100]

$$
\begin{aligned}
\bar{\Omega}^{\langle 2\rangle} =& \Bigg\{\boldsymbol{\omega} : \bigwedge_{n_\mathrm{P}=1}^{N_\mathrm{P}} \Bigg(\bigvee_{\bar{h}=1}^{h} P_{\bar{h},n_\mathrm{P}} = 0 \\
& \vee \bigvee_{\bar{h}_1=1}^{h-1} \bigvee_{\bar{h}_2 \in \bar{h}} \left(P_{\bar{h}_1,n_\mathrm{P}} = P_{\bar{h}_2,n_\mathrm{P}} \vee P_{\bar{h}_1,n_\mathrm{P}} = -P_{\bar{h}_2,n_\mathrm{P}}\right) \\
& \vee \bigvee_{\substack{\bar{h}_{1/2}=1 \\ \bar{h}_1=\bar{h}_2}}^{h} \bigvee_{\bar{l}\in\bar{L}} \left(P_{\bar{h}_1,n_\mathrm{P}} = \underline{\omega}_{\bar{l},\bar{h}_2} \vee P_{\bar{h}_1,n_\mathrm{P}} = -\underline{\omega}_{\bar{l},\bar{h}_2}\right) \\
& \vee \bigvee_{\substack{\bar{h}_{1/2}=1 \\ \bar{h}_1=\bar{h}_2}}^{h} \bigvee_{\bar{k}\in\bar{K}} \left(P_{\bar{h}_1,n_\mathrm{P}} = \underline{\bar{\omega}}_{\bar{k},\bar{h}_2} \vee P_{\bar{h}_1,n_\mathrm{P}} = -\underline{\bar{\omega}}_{\bar{k},\bar{h}_2}\right) \Bigg)\Bigg\} \qquad (5.45a)\\
=& \Bigg\{\boldsymbol{\omega} : \underbrace{\left(\bigvee (\cdot) = (\cdot)\right)\Big|_{n_\mathrm{P}=1} \wedge \left(\bigvee (\cdot) = (\cdot)\right)\Big|_{n_\mathrm{P}=2}}_{\mathrm{IV}} \wedge \bigwedge_{n_\mathrm{P}=3}^{N_\mathrm{P}} \left(\bigvee (\cdot) = (\cdot)\right) \Bigg\} \qquad (5.45b)\\
=& \Bigg\{\boldsymbol{\omega} : \left(\bigvee_{z} \boldsymbol{C}_z \boldsymbol{\omega} = \boldsymbol{0}\right) \wedge \bigwedge_{n_\mathrm{P}=3}^{N_\mathrm{P}} \left(\bigvee (\cdot) = (\cdot)\right) \Bigg\} \qquad (5.45c)\\
=& \Bigg\{\boldsymbol{\omega} : \bigvee_{z=1}^{Z_2} \boldsymbol{C}_z \boldsymbol{\omega} = \boldsymbol{0} \Bigg\}. \qquad (5.45d)
\end{aligned}
$$

Für den Spezialfall, dass im letzten Vereinfachungsschritt von $\bar{\Omega}^{\langle 2\rangle}$, d. h. für $k_\mathrm{it} \geq N_\mathrm{P} - 1$, $\omega_j = \omega_j, \forall j \in \{1, \ldots, m\}$, folgt, ist die Aussage $\bar{\Omega}^{\langle 2\rangle}$ für beliebige $\boldsymbol{\omega} \in \mathbb{R}^m$ stets wahr und es resultiert $\bar{\Omega}^{\langle 2\rangle} = \mathbb{R}^m$ und somit $\Omega^{\langle 2\rangle} = \emptyset$. Andernfalls ist $\bar{\Omega}^{\langle 2\rangle}$ durch die Disjunktion von $Z_2 \in \mathbb{N}_{\geq 1}$ Bedingungen der Form $\boldsymbol{C}_z\boldsymbol{\omega} = \boldsymbol{0}$, $z = 1, 2, \ldots, Z_2$, gegeben (vgl. (5.45d)). Dies wird im nachfolgenden Lemma formal gezeigt.

[100] Für den Spezialfall $N_\mathrm{P} = 1$ entfällt die Konjunktion mit den durch $n_\mathrm{P} = k_\mathrm{it} + 1$ indizierten Bedingungen in Schritt 1 von Algorithmus 5.2 und (5.45d) folgt aus (5.45a).

Algorithmus 5.2 Vereinfachung von $\bar{\Omega}^{\langle 2\rangle}$

Initialisiere die Iterationsvariable $k_{\mathrm{it}} := 1$.

Schritt 1

Aufstellen homogener linearer Gleichungssysteme, indem die Konjunktion der mit $n_{\mathrm{P}} = k_{\mathrm{it}}$ und $n_{\mathrm{P}} = k_{\mathrm{it}} + 1$ indizierten Bedingungen unter Anwendung des Distributivgesetzes expandiert wird. Falls $k_{\mathrm{it}} + 1 > N_{\mathrm{P}}$ gilt, entfällt diese Konjunktion.

Schritt 2

Lösen der linearen Gleichungssysteme nach $\boldsymbol{\omega}$. Hierbei werden frei wählbare Variablen zu $\omega_j = \omega_j$ gesetzt und die Bedingungen durch $\bar{\boldsymbol{C}}_z\boldsymbol{\omega} = \mathbf{0}$ ($\bar{\boldsymbol{C}}_z \in \mathbb{R}^{m\times m}$) mit dem Laufindex z ausgedrückt.

if $\omega_j = \omega_j$ für jedes $j \in \{1, \ldots, m\}$ **then**
    if $k_{\mathrm{it}} \geq N_{\mathrm{P}} - 1$ **then**
        return $\bar{\Omega}^{\langle 2\rangle} = \mathbb{R}^m$
    else
        Setze $k_{\mathrm{it}} := k_{\mathrm{it}} + 2$ und gehe zu Schritt 1.
    end if
else
    Setze $k_{\mathrm{it}} := k_{\mathrm{it}} + 1$ und gehe zu Schritt 3.
end if

Schritt 3

Transformation der Koeffizientenmatrizen $\bar{\boldsymbol{C}}_z$ auf reduzierte Zeilenstufenform, Entfernen aller Nullzeilen. Resultierende Matrizen werden mit $\boldsymbol{C}_z$ ($\boldsymbol{C}_z \in \mathbb{R}^{\bar{m}\times m}, \bar{m} < m$) bezeichnet.

Schritt 4

Entfernen mehrfach auftretender Bedingungen (identischer $\boldsymbol{C}_z$). Alle bisher expandierten Konjunktionen werden anschließend zu $\bigvee_z \boldsymbol{C}_z\boldsymbol{\omega} = \mathbf{0}$ zusammengefasst.

Schritt 5

while $k_{\mathrm{it}} \leq N_{\mathrm{P}} - 1$ **do**
    Stelle lineare homogene Gleichungssysteme durch die Expansion der Konjunktion zwischen dem Ergebnis aus Schritt 4 und den durch $n_{\mathrm{P}} = k_{\mathrm{it}} + 1$ indizierten Bedingungen auf. Wende die Schritte 2–4 auf diese Gleichungssysteme an.
end while
return $\bar{\Omega}^{\langle 2\rangle}$

Lemma 5.5 (Vereinfachung der Menge $\Omega^{\langle 2\rangle}$)

Falls Algorithmus 5.2 die Menge $\bar{\Omega}^{\langle 2\rangle} = \mathbb{R}^m$ ausgibt, so folgt $\Omega^{\langle 2\rangle} = \emptyset$. Andernfalls ist (5.43) *identisch zu*

$$\Omega^{\langle 2\rangle} = \left\{\boldsymbol{\omega} : \bigwedge_{z=1}^{Z_2} \boldsymbol{C}_z\boldsymbol{\omega} \neq \mathbf{0}\right\} \tag{5.46}$$

mit $Z_2 \in \mathbb{N}_{\geq 1}$.

Beweis:
Um zu beweisen, dass (5.46) dieselbe Menge wie (5.43) definiert, muss sowohl die Gültigkeit der Vereinfachungsschritte 3 und 4 nach Algorithmus 5.2 als auch die Korrektheit der Schlussfolgerungen im Fall $\omega_j = \omega_j, \forall j \in \{1, \dots, m\}$, in Schritt 2 betrachtet werden.

Zunächst wird Schritt 3 analysiert. Die Transformation der Koeffizientenmatrix eines homogenen linearen Gleichungssystems auf reduzierte Zeilenstufenform verändert dessen Lösungsmenge nicht. Des Weiteren entsprechen Nullzeilen in $\bar{\boldsymbol{C}}_z$ dem Fall $\omega_j = \omega_j$ und stellen somit Bedingungen dar, die durch $\Omega^{\langle 2 \rangle}$ nicht erfüllt werden können. Folglich können Nullzeilen in $\bar{\boldsymbol{C}}_z$ entfernt werden. Schritt 4 ist gerechtfertigt, da mehrfach auftretende Bedingungen keine weiteren Informationen hinsichtlich der Definition der gesuchten Menge beinhalten und somit redundant sind.

Nun werde o. B. d. A. der Fall betrachtet, in dem sich in Schritt 2 für die durch IV gekennzeichneten Bedingungen in (5.45) $\omega_j = \omega_j$ für alle $j \in \{1, \dots, m\}$ ergibt. Dann ist Teil IV für jedes beliebige $\boldsymbol{\omega} \in \mathbb{R}^m$ unabhängig von anderen Bedingungen der Form $\bar{\boldsymbol{C}}_z \boldsymbol{\omega} = \boldsymbol{0}$ in IV wahr. Somit können diese anderen Bedingungen vernachlässigt werden und Schritt 1 muss mit $n_\mathrm{P} = 3$ und $n_\mathrm{P} = 4$ ausgeführt werden. Da sich diese Überlegungen ebenfalls auf die Wiederholungen in Schritt 5 übertragen lassen und der Spezialfall, dass $\omega_j = \omega_j$, $\forall j \in \{1, \dots, m\}$, in der letzten Iteration $k_\mathrm{it} \geq N_\mathrm{P} - 1$ resultiert, explizit berücksichtigt wird, folgt die Gültigkeit von Lemma 5.5. □

5.4.3 Frequenzbedingungen Ω

Die aus dem Schnitt der Hilfsmengen $\Omega^{\langle 1 \rangle}$ und $\Omega^{\langle 2 \rangle}$ resultierenden Frequenzbedingungen Ω lassen sich, wie im folgenden Lemma gezeigt, weiter vereinfachen.

Lemma 5.6 (Frequenzbedingungen Ω)
Die Schnittmenge zwischen $\Omega^{\langle 1 \rangle}$ nach (5.38) *und $\Omega^{\langle 2 \rangle}$ nach* (5.46) *mit $Z_2 \in \mathbb{N}_{\geq 1}$ ergibt*

$$\begin{aligned} \Omega = \Omega^{\langle 1 \rangle} \cap \Omega^{\langle 2 \rangle} &= \left\{ \boldsymbol{\omega} : \bigwedge_{z=1}^{Z_1 + Z_2} \boldsymbol{C}_z \boldsymbol{\omega} \neq \boldsymbol{0} \right\} \\ &= \left\{ \boldsymbol{\omega} : \bigwedge_{z=1}^{Z} \boldsymbol{C}_z \boldsymbol{\omega} \neq \boldsymbol{0} \right\}, \quad Z \in \mathbb{N}_{\geq 1}, \end{aligned} \tag{5.47}$$

mit $Z \leq Z_1 + Z_2$. Die letzte Gleichung resultiert, indem jede Matrix $\boldsymbol{C}_{z_2}$ entfernt wird, für die bereits eine andere Matrix $\boldsymbol{C}_{z_1}$, $z_1 \neq z_2$, in der Konjunktion (5.47) *enthalten ist und außerdem $\boldsymbol{M} \in \mathbb{R}^{m_1 \times m_2}$ mit $m_1 \leq m_2$ existiert, sodass $\boldsymbol{C}_{z_1} = \boldsymbol{M} \boldsymbol{C}_{z_2}$ gilt.*

Beweis:
Die Schnittmenge $\Omega = \Omega^{\langle 1 \rangle} \cap \Omega^{\langle 2 \rangle}$ ergibt $Z_1 + Z_2$ Bedingungen, die durch die Koeffizientenmatrizen $\boldsymbol{C}_z$ formuliert sind[101]. Um die Anzahl der Bedingungen zu reduzieren, kann jede Matrix $\boldsymbol{C}_{z_2}$ vernachlässigt werden, deren Bedingungen bereits durch die Matrix $\boldsymbol{C}_{z_1} = \boldsymbol{M}\boldsymbol{C}_{z_2}$, $z_1 \neq z_2$, sichergestellt sind. Diese Redundanz der Matrix $\boldsymbol{C}_{z_2}$ gilt, da aus $\boldsymbol{C}_{z_1}\boldsymbol{\omega} \neq \boldsymbol{0} \Leftrightarrow \boldsymbol{M}\boldsymbol{C}_{z_2}\boldsymbol{\omega} \neq \boldsymbol{0}$ direkt $\boldsymbol{C}_{z_2}\boldsymbol{\omega} \neq \boldsymbol{0}$ folgt. □

Nach diesen Vorbereitungen kann schließlich die Hauptaussage dieses Kapitels formuliert werden, die belegt, dass die Menge Ω, welche Bedingungen an die Frequenzen $\boldsymbol{\omega}$ in $\boldsymbol{x}_{\mathrm{PE}}$ formuliert, eine Menge im Sinne von Proposition 5.1 darstellt.

Satz 5.2 (PE von $\boldsymbol{\sigma} = \dot{\boldsymbol{\phi}}(\boldsymbol{x}_{\mathrm{PE}})\ \forall \boldsymbol{\omega} \in \Omega$)
Seien $\boldsymbol{\phi}(\boldsymbol{x})$ und $\boldsymbol{x}_{\mathrm{PE}}$ entsprechend Annahme 5.3 und Annahme 5.4 gewählt und Ω wie in Lemma 5.6 definiert. Falls $\Omega \neq \emptyset$ gilt, so garantiert jeder beliebige Vektor $\boldsymbol{\omega} \in \Omega$, dass das Signal $\boldsymbol{x}_{\mathrm{PE}}(t)$ SR bezüglich $\dot{\boldsymbol{\phi}}(\cdot)$ ist und somit $\boldsymbol{\sigma} = \dot{\boldsymbol{\phi}}(\boldsymbol{x}_{\mathrm{PE}})$ PE ist.

Beweis:
Nach Lemma 5.2 lässt sich $\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{PE}})$ durch (5.31) ausdrücken. Darauf basierend lassen sich die Hilfsmengen $\Omega^{\langle 1 \rangle}$ nach (5.37) und $\Omega^{\langle 2 \rangle}$ nach (5.43) definieren. Da Lemma 5.4, Lemma 5.5 und Lemma 5.6 lediglich Äquivalenztransformationen anwenden und die Mengen $\Omega^{\langle 1 \rangle}$, $\Omega^{\langle 2 \rangle}$ und Ω nicht verändern, muss im Folgenden nur gezeigt werden, dass die Bedingungen in (5.37) und (5.43) hinreichend sind, damit (5.35) gilt. Dann folgt nach Proposition 5.1, dass $\boldsymbol{x}_{\mathrm{PE}}(t)$ für jeden Vektor $\boldsymbol{\omega} \in \Omega$ ein SR-Signal bezüglich $\dot{\boldsymbol{\phi}}(\cdot)$ darstellt und somit $\boldsymbol{\sigma} = \dot{\boldsymbol{\phi}}(\boldsymbol{x}_{\mathrm{PE}})$ PE ist.

Die zeitliche Ableitung von (5.31) liefert

$$\sigma_{\bar{h}} = \sum_{l=1}^{L_{\bar{h}}} a_{l,\bar{h}} \underline{\omega}_{l,\bar{h}} \cos\big(\underline{\omega}_{l,\bar{h}} t\big) - \sum_{k=1}^{K_{\bar{h}}} c_{k,\bar{h}} \bar{\omega}_{k,\bar{h}} \sin\big(\bar{\omega}_{k,\bar{h}} t\big), \tag{5.48}$$

$\forall \bar{h} \in \{1, \dots, h\}$. Mit $\boldsymbol{\alpha} \in \mathbb{R}^h$ folgt aus (5.48)

$$\begin{aligned} \boldsymbol{\alpha}^\intercal \boldsymbol{\sigma} = \quad & \alpha_1 \left(\sum_{l=1}^{L_1} a_{l,1} \underline{\omega}_{l,1} \cos\big(\underline{\omega}_{l,1} t\big) - \sum_{k=1}^{K_1} c_{k,1} \bar{\omega}_{k,1} \sin\big(\bar{\omega}_{k,1} t\big) \right) \\ & + \dots \\ & + \alpha_h \left(\sum_{l=1}^{L_h} a_{l,h} \underline{\omega}_{l,h} \cos\big(\underline{\omega}_{l,h} t\big) - \sum_{k=1}^{K_h} c_{k,h} \bar{\omega}_{k,h} \sin\big(\bar{\omega}_{k,h} t\big) \right) \\ = & \sum_{\bar{h}=1}^{h} \left(\alpha_{\bar{h}} \left(\sum_{l=1}^{L_{\bar{h}}} a_{l,\bar{h}} \underline{\omega}_{l,\bar{h}} \cos\big(\underline{\omega}_{l,\bar{h}} t\big) \ - \sum_{k=1}^{K_{\bar{h}}} c_{k,\bar{h}} \bar{\omega}_{k,\bar{h}} \sin\big(\bar{\omega}_{k,\bar{h}} t\big) \right) \right). \end{aligned} \tag{5.49}$$

[101] Hierbei werden die Zeilenvektoren $\boldsymbol{c}_z^\intercal$ in (5.38) ebenfalls als Koeffizientenmatrizen $\boldsymbol{C}_z$ aufgefasst.

O. B. d. A. werde nun für die beispielhaft betrachtete (L_h+1)-te Spalte

$$\mathcal{P}_{L_h+1} = \left(\zeta_1^{\langle L_h+1\rangle}, \dots, \zeta_h^{\langle L_h+1\rangle}\right) \tag{5.50}$$

von $\boldsymbol{P}^{\mathrm{f}}$ in (5.41) mit $\zeta_{\bar{h}}^{\langle L_h+1\rangle} = \underline{\omega}_{1,\bar{h}}, \forall \bar{h} \in \{1,\dots,h-1\}$, und $\zeta_h^{\langle L_h+1\rangle} = \bar{\underline{\omega}}_{1,h}$ angenommen, dass die Bedingungen I, II und III in (5.43) erfüllt sind. Ausgehend von (5.49) werden die trigonometrischen Funktionen, welche die entsprechenden Frequenzen dieser Spalte als Argumente beinhalten, separiert, und die verbleibenden trigonometrischen Ausdrücke innerhalb der Klammern $\alpha_{\bar{h}}(\cdot), \forall \bar{h} \in \{1,\dots,h\}$, zu $\varepsilon_{\bar{h}}$ zusammengefasst. Damit folgt

$$\begin{aligned}\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\sigma} = \quad & \alpha_1\big(a_{1,1}\underline{\omega}_{1,1}\cos\big(\underline{\omega}_{1,1}t\big) + \varepsilon_1\big) \\ & + \dots \\ & + \alpha_{h-1}\big(a_{1,h-1}\underline{\omega}_{1,h-1}\cos\big(\underline{\omega}_{1,h-1}t\big) + \varepsilon_{h-1}\big) + \alpha_h\big(-c_{1,h}\bar{\underline{\omega}}_{1,h}\sin\big(\bar{\underline{\omega}}_{1,h}t\big) + \varepsilon_h\big) \\ = & \sum_{\bar{h}=1}^{h-1}\alpha_{\bar{h}}a_{1,\bar{h}}\underline{\omega}_{1,\bar{h}}\cos\big(\underline{\omega}_{1,\bar{h}}t\big) - \alpha_h c_{1,h}\bar{\underline{\omega}}_{1,h}\sin\big(\bar{\underline{\omega}}_{1,h}t\big) + \sum_{\bar{h}=1}^{h}\alpha_{\bar{h}}\varepsilon_{\bar{h}}.\end{aligned} \tag{5.51}$$

Die Bedingungen in $\Omega^{\langle 1\rangle}$ nach (5.37) stellen sicher, dass die separierten trigonometrischen Funktionen in (5.51) innerhalb der Klammern $\alpha_{\bar{h}}(\cdot)$ nicht durch trigonometrische Ausdrücke derselben Art eliminiert werden. Dies folgt aus der geforderten Einzigartigkeit der Frequenzen in jedem Element $\phi_{\bar{h}}(\boldsymbol{x}_{\mathrm{PE}})$ gemäß (5.31) (siehe Definition 5.6). Einzigartigkeit meint hierbei, dass Frequenzen, die Argumente gleichartiger trigonometrischer Funktionen sind, ungleich zueinander und zu ihrer negierten Frequenz sind. Darüber hinaus garantieren die Bedingungen in $\Omega^{\langle 1\rangle}$, dass in dem Element $\phi_{\bar{h}}(\boldsymbol{x}_{\mathrm{PE}})$ $L_{\bar{h}}$ Sinus- und $K_{\bar{h}}$ Kosinusfunktionen existent sind und zwei Sinus- bzw. Kosinusfunktionen sich nicht auf eine einzelne Sinus- bzw. Kosinusfunktion reduzieren lassen. Hierdurch wird gewährleistet, dass die Kardinalität jeder Menge $W_{\bar{h}}$, $\bar{h} \in \{1,\dots,h\}$, (vgl. (5.39)) $L_{\bar{h}} + K_{\bar{h}}$ entspricht. Dies ist relevant, damit die Matrix $\boldsymbol{P}^{\mathrm{f}}$ der Frequenztupel nach Definition 5.7 aufgestellt werden kann und Bedingungen an die Spalten von $\boldsymbol{P}^{\mathrm{f}}$ abgeleitet werden können.

Die durch III gekennzeichneten Bedingungen in $\Omega^{\langle 2\rangle}$ (5.43) gewährleisten, dass die trigonometrische Funktion, die innerhalb der Klammern $\alpha_{\bar{h}_1}(\cdot)$ in (5.51) separiert wurde, nicht durch trigonometrische Funktionen in $\varepsilon_{\bar{h}_2}$ innerhalb der Klammer $\alpha_{\bar{h}_2}(\cdot)$ $(\forall \bar{h}_1, \bar{h}_2 \in \{1,\dots,h\}, \bar{h}_1 \neq \bar{h}_2)$ eliminiert wird. Auch dies gilt wegen der Einzigartigkeit der auftretenden Frequenzen.

Zusammenfassend können aufgrund der Bedingungen in $\Omega^{\langle 1\rangle}$ und der Bedingungen III in $\Omega^{\langle 2\rangle}$ die separierten trigonometrischen Ausdrücke in (5.51), unabhängig von den konkreten Werten von $\alpha_{\bar{h}}$, $a_{l,\bar{h}}$ und $c_{k,\bar{h}}$, nicht durch die Funktionen in $\varepsilon_{\bar{h}}, \forall \bar{h} \in \{1,\dots,h\}$, eliminiert werden. Daher genügt es, zu zeigen, dass die Bedingungen, die durch I und II in $\Omega^{\langle 2\rangle}$ (5.43) gekennzeichnet sind, unter der Annahme $\varepsilon_{\bar{h}} = 0, \forall \bar{h} \in \{1,\dots,h\}$, zu einem Signal $\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\sigma}$ nach (5.35) führen. Falls $\varepsilon_{\bar{h}} \neq 0$ gilt, können die in (5.51) separierten trigonometrischen Funktionen nicht ausgelöscht werden, es können höchstens zusätzliche Terme hinzukommen.

Im Folgenden gelte $\varepsilon_{\bar{h}} = 0, \forall \bar{h} \in \{1,\dots,h\}$. Aufgrund der Bedingungen I in $\Omega^{\langle 2\rangle}$ verschwinden die Frequenzen der trigonometrischen Ausdrücke, die in (5.51) separiert wurden, nicht und

es folgt $\underline{\omega}_{1,\bar{h}} \neq 0$, $a_{1,\bar{h}}\underline{\omega}_{1,\bar{h}} \neq 0, \forall \bar{h} \in \{1,\dots,h-1\}$, sowie $\bar{\underline{\omega}}_{1,h} \neq 0$ und $c_{1,h}\bar{\underline{\omega}}_{1,h} \neq 0$. Falls $\varepsilon_{\bar{h}} = 0\ \forall \bar{h} \in \{1,\dots,h\}$ gilt, sind diese Bedingungen notwendig für (5.35). Dies lässt sich wie folgt einsehen. Wäre eine Frequenz null, so existierte ein $\boldsymbol{\alpha} \neq \mathbf{0}$, sodass $\boldsymbol{\alpha}^\mathsf{T}\boldsymbol{\sigma} = 0$ folgt. Würde beispielsweise das $\bar{h}$-te Element von $\boldsymbol{\sigma}$ die verschwindende Frequenz beinhalten, so wäre ein solches $\boldsymbol{\alpha}$ gegeben, indem alle Einträge von $\boldsymbol{\alpha}$ außer dem $\bar{h}$-ten zu null gesetzt würden. Mit den durch II gekennzeichneten Bedingungen in $\Omega^{\langle 2 \rangle}$ bilden die separierten trigonometrischen Ausdrücke $a_{1,1}\underline{\omega}_{1,1}\cos\big(\underline{\omega}_{1,1}t\big), \dots, a_{1,h-1}\underline{\omega}_{1,h-1}\cos\big(\underline{\omega}_{1,h-1}t\big), c_{1,h}\bar{\underline{\omega}}_{1,h}\sin\big(\bar{\underline{\omega}}_{1,h}t\big)$ in (5.51) ein linear unabhängiges Funktionensystem. Somit erfüllt ihre Linearkombination (5.35), falls mindestens einer der Koeffizienten ungleich null ist, also $\boldsymbol{\alpha} \neq \mathbf{0}$ gilt. Im Spezialfall $\varepsilon_{\bar{h}} = 0$ $\forall \bar{h} \in \{1,\dots,h\}$ sind die Bedingungen II in $\Omega^{\langle 2 \rangle}$ wiederum ebenfalls notwendig zur Erfüllung von (5.35). Um dies einzusehen, sei o. B. d. A. $\underline{\omega}_{1,1} = \pm\underline{\omega}_{1,h-1}$. Dann folgt $\boldsymbol{\alpha}^\mathsf{T}\boldsymbol{\sigma} = 0$ für $\boldsymbol{\alpha}^\mathsf{T} = \begin{bmatrix} \mp\frac{1}{a_{1,1}} & 0 & \dots & 0 & \frac{1}{a_{1,h-1}} & 0 \end{bmatrix}$.

Somit sind für den Fall $\varepsilon_{\bar{h}} = 0$ die Bedingungen I und II in $\Omega^{\langle 2 \rangle}$ notwendig und hinreichend dafür, dass (5.35) für jedes beliebige $\boldsymbol{\alpha} \neq \mathbf{0}$ gilt. Wie zuvor gezeigt, sind die zusätzlichen Bedingungen im Fall $\varepsilon_{\bar{h}} \neq 0$ hinreichend dafür, dass sich trigonometrische Funktionen nicht aufheben. Da die beispielhaft gewählte Frequenzkombination, d. h. die Spalte aus $\boldsymbol{P}^\mathrm{f}$ im hier erfolgten Beweis, austauschbar ist und mindestens eine solche Spalte existiert, führt jeder Frequenzvektor $\boldsymbol{\omega} \in \Omega = \Omega^{\langle 1 \rangle} \cap \Omega^{\langle 2 \rangle}$ zu einem Signal $\boldsymbol{\alpha}^\mathsf{T}\boldsymbol{\sigma}$, für das (5.35) gilt. Schlussendlich folgt aus Proposition 5.1, dass $\boldsymbol{x}_\mathrm{PE}$ SR bezüglich der Transformation $\dot{\boldsymbol{\phi}}(\cdot)$ ist und somit $\boldsymbol{\sigma}$ die PE-Eigenschaft erfüllt. □

Satz 5.2 liefert ein hinreichendes Kriterium für die Wahl der Frequenzen $\boldsymbol{\omega}$ in $\boldsymbol{x}_\mathrm{PE}$, sodass das Signal $\boldsymbol{x}_\mathrm{PE}$ SR bezüglich $\dot{\boldsymbol{\phi}}(\cdot)$ ist. Das folgende Lemma adressiert die Annahme in Satz 5.2, dass Ω eine nichtleere Menge ist.

Lemma 5.7

Falls der sukzessive Vereinfachungsprozess von $\bar{\Omega}^{\langle 2 \rangle}$ nach Algorithmus 5.2 in der letzten Iteration ($k_\mathrm{it} \geq N_\mathrm{P} - 1$) nicht zur allgemeinen Lösung $\omega_j = \omega_j, \forall j \in \{1,\dots,m\}$, führt, ergibt sich Ω wie in (5.47) *und es gilt $\Omega \neq \emptyset$.*

Beweis:

Tritt $\omega_j = \omega_j, \forall j \in \{1,\dots,m\}$, in der letzten Iteration $k_\mathrm{it} \geq N_\mathrm{P} - 1$ von $\bar{\Omega}^{\langle 2 \rangle}$ in Algorithmus 5.2 nicht auf, so folgt nach Lemma 5.5 die nichtleere Menge $\Omega^{\langle 2 \rangle}$ durch (5.46). Zudem ist nach Lemma 5.6 Ω durch (5.47) gegeben. Unter der Annahme, jede Zeile $\boldsymbol{c}_z^\mathsf{T}$ der Koeffizientenmatrizen $\boldsymbol{C}_z$ erfülle die Ungleichung $\boldsymbol{c}_z^\mathsf{T}\boldsymbol{\omega} \neq 0$, lässt sich eine Teilmenge Ω_I von Ω definieren, wobei die durch Ω_I beschriebenen Bedingungen einschränkender als die Bedingungen in Ω sind, d. h.

$$\Omega_\mathrm{I} = \left\{ \boldsymbol{\omega} : \bigwedge_{z} \boldsymbol{c}_z^\mathsf{T}\boldsymbol{\omega} \neq 0 \right\} \subset \Omega = \left\{ \boldsymbol{\omega} : \bigwedge_{z=1}^{Z} \boldsymbol{C}_z\boldsymbol{\omega} \neq \mathbf{0} \right\}. \tag{5.52}$$

Da $\boldsymbol{c}_z^\intercal \boldsymbol{\omega} = 0$ eine $(m-1)$-dimensionale Hyperebene im $\mathbb{R}^m$ beschreibt, ist deren Lebesgue-Maß null und die Hyperebene definiert eine Lebesguesche Nullmenge [Coh13, S. 146]. Zudem führt jede abzählbare Vereinigung Lebesguescher Nullmengen wieder auf eine Lebesguesche Nullmenge [Coh13, Proposition 1.2.4]. Daraus folgt, dass Ω_I den gesamten Raum $\mathbb{R}^m$ mit Ausnahme einer Lebesgueschen Nullmenge enthält und somit $\Omega_\mathrm{I} \neq \emptyset$ sowie $\Omega \neq \emptyset$ gilt. □

Bemerkung 5.3

Sollte in der letzten Iteration von $\bar{\Omega}^{\langle 2 \rangle}$ nach Algorithmus 5.2 $\omega_j = \omega_j, \forall j \in \{1, \ldots, m\}$, resultieren, d. h. sollten für die Komplementärmenge $\bar{\Omega}^{\langle 2 \rangle}$ keinerlei Einschränkungen an die Wahl der Frequenzen gelten, so ist es ratsam, zusätzliche Frequenzvariablen in den Elementen der Anregungstrajektorie $\boldsymbol{x}_\mathrm{PE}$ einzuführen, bis eine nichtleere Menge Ω resultiert.

Für den Spezialfall $N_\mathrm{P} = 1$, d. h., falls $\boldsymbol{P}^\mathrm{f}$ nach Definition 5.7 nur eine Spalte besitzt, sind die Frequenzbedingungen aus Satz 5.2 notwendig und hinreichend. Zudem bleibt die PE-Eigenschaft von $\boldsymbol{\sigma} = \dot{\boldsymbol{\phi}}(\boldsymbol{x}_\mathrm{PE})$ bei einer Skalierung von $\boldsymbol{x}_\mathrm{PE}$ erhalten. Diese Aussagen sind in Lemma 5.8 bzw. Lemma 5.9 formalisiert.

Lemma 5.8

Seien $\boldsymbol{x}_\mathrm{PE}$ und $\boldsymbol{\phi}(\boldsymbol{x})$ wie in Satz 5.2 gewählt und Ω wie in Lemma 5.6 gegeben. Falls $N_\mathrm{P} = 1$ gilt, ist $\boldsymbol{\sigma} = \dot{\boldsymbol{\phi}}(\boldsymbol{x}_\mathrm{PE})$ genau dann PE, wenn $\boldsymbol{\omega} \in \Omega$.

Beweis:

Für $N_\mathrm{P} = 1$ ergibt sich $\Omega^{\langle 1 \rangle} = \mathbb{R}^m$. Da hierbei zudem die durch III in $\Omega^{\langle 2 \rangle}$ (5.43) gekennzeichneten Bedingungen entfallen, spielen lediglich die Teilbedingungen I und II von $\Omega^{\langle 2 \rangle}$ bei der Definition von Ω eine Rolle. Außerdem impliziert $N_\mathrm{P} = 1$ in (5.51) $\varepsilon_{\bar{h}} = 0, \forall \bar{h} \in \{1, \ldots, h\}$. Für diesen Fall sind gemäß dem Beweis von Satz 5.2 die Bedingungen I und II in $\Omega^{\langle 2 \rangle}$ notwendig und hinreichend dafür, dass (5.35), aber insbesondere auch $\boldsymbol{\alpha}^\intercal \boldsymbol{\sigma} \neq 0$, für jedes beliebige $\boldsymbol{\alpha} \neq \boldsymbol{0}$ gilt. Mit Lemma 5.3 folgt schließlich die gesuchte Aussage. □

Lemma 5.9 (Skalierung von $\boldsymbol{x}_\mathrm{PE}$)

Seien $\boldsymbol{x}_\mathrm{PE}$ und Ω entsprechend Satz 5.2 gegeben. Dann ist

$$\bar{\boldsymbol{x}}_\mathrm{PE} = \begin{bmatrix} \nu_1 & 0 & \ldots & 0 \\ 0 & \nu_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & \nu_n \end{bmatrix} \boldsymbol{x}_\mathrm{PE} \tag{5.53}$$

für jeden Vektor $\boldsymbol{\omega} \in \Omega$ und $\nu_j \in \mathbb{R}_{\neq 0}, \forall j \in \{1, \ldots, n\}$, ein SR-Signal bezüglich $\dot{\boldsymbol{\phi}}(\cdot)$.

Beweis:
Siehe Anhang C.2. □

Während zu Beginn von Abschnitt 5.4 der Spielerindex i zunächst vernachlässigt wurde, sollen nun wieder alle N Spieler berücksichtigt werden. Dabei können sich deren Basisfunktionsvektoren $\boldsymbol{\phi}_i(\boldsymbol{x})$, $i \in \mathcal{N}$, im Allgemeinen unterscheiden. Für ein gegebenes $\boldsymbol{x}_{\mathrm{PE}}$ nach Annahme 5.4 wird für jeden der N Basisfunktionsvektoren $\boldsymbol{\phi}_i(\boldsymbol{x})$ nach Annahme 5.3 eine Menge an Frequenzbedingungen nach Satz 5.2 berechnet. Daraus ergeben sich N Mengen Ω_i. Wird deren Schnittmenge analog zu Lemma 5.6 gebildet, folgt die Menge

$$\Omega = \bigcap_{i=1}^{N} \Omega_i. \tag{5.54}$$

Jeder Vektor $\boldsymbol{\omega} \in \Omega$ stellt dann sicher, dass $\boldsymbol{x}_{\mathrm{PE}}(t)$ SR bezüglich $\dot{\boldsymbol{\phi}}_i(\cdot)$, $\forall i \in \mathcal{N}$, ist.

Satz 5.2 stellt das Hauptergebnis dieses Kapitels dar und liefert eine neuartige und generell anwendbare Aussage bezüglich PE-Signalen, die aus nichtlinearen, polynomiellen Transformationen resultieren. Die durch die Menge Ω definierten Frequenzbedingungen können für ADP-Systeme angewandt werden, welche die Erfüllung der PE-Bedingung nach Definition 5.4 erfordern. Insbesondere liefert Satz 5.2 eine Lösung für Problem 5.1, da $\boldsymbol{x}_{\mathrm{PE}}(t)$ mit beliebigem $\boldsymbol{\omega} \in \Omega$ ein $\boldsymbol{x}(t)$ im Sinne von Problem 5.1 darstellt. Da aus praktischer Sicht die Überprüfung, ob die PE-Eigenschaft bei vorliegenden Signalen tatsächlich erfüllt ist, bislang nur unzureichend geklärt ist, beschäftigt sich der nächste Abschnitt mit dieser Fragestellung.

5.5 Signal zur Überprüfung der Erfüllung der PE-Eigenschaft

Im Folgenden werden zwei Eigenwertsignale vorgestellt, mit deren Hilfe eine Antwort auf die Frage, ob ein vorliegendes Signal die PE-Eigenschaft erfüllt, gegeben werden kann.

Lemma 5.10 (Eigenwertsignal $\lambda_{\min,1}(t)$ zur Überprüfung der PE-Eigenschaft)
Das Signal $\boldsymbol{\sigma}(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^h$ ist genau dann PE $\forall t \geq t_0$, wenn Konstanten $\alpha > 0$ und $T > 0$ existieren, sodass $\forall t \geq t_0$

$$\lambda_{\min,1}(t) \coloneqq \lambda_{\min}\left(\boldsymbol{\Xi}_1(t)\right) \coloneqq \lambda_{\min}\left(\int_t^{t+T} \boldsymbol{\sigma}(\tau)\boldsymbol{\sigma}^{\mathsf{T}}(\tau)\,\mathrm{d}\tau\right) \geq \alpha > 0 \tag{5.55}$$

gilt. Hierbei bezeichnet $\lambda_{\min}(\cdot)$ den kleinsten Eigenwert einer Matrix[102].

[102] Da es sich bei $\boldsymbol{\Xi}_1(t)$ um eine symmetrische Matrix handelt, sind deren Eigenwerte alle reell und der kleinste Eigenwert ist eindeutig definiert.

Beweis:
Nach Definition 5.4 ist $\boldsymbol{\sigma}(t)$ genau dann PE, wenn

$$\Xi_1(t) := \int_t^{t+T} \boldsymbol{\sigma}(\tau)\boldsymbol{\sigma}^\mathsf{T}(\tau)\,\mathrm{d}\tau \succeq \alpha \boldsymbol{I} \tag{5.56}$$

gilt. Da $\Xi_1(t)$ nach (5.56) $\forall t$ symmetrisch ist, existiert eine reguläre Matrix $\boldsymbol{M}(t)$, sodass

$$\Xi_1(t) = \boldsymbol{M}(t)\boldsymbol{G}(t)\boldsymbol{M}^{-1}(t) \tag{5.57}$$

gilt, wobei $\boldsymbol{G}(t)$ eine Diagonalmatrix mit den Eigenwerten von $\Xi_1(t)$ ist [BSMM13, S. 325]. Die PE-Eigenschaft von $\boldsymbol{\sigma}(t)$ ist folglich äquivalent zu

$$\Xi_1(t) = \boldsymbol{M}(t)\boldsymbol{G}(t)\boldsymbol{M}^{-1}(t) \succeq \alpha \boldsymbol{I} \tag{5.58}$$

und aufgrund der Regularität von $\boldsymbol{M}(t)$ zu

$$\boldsymbol{G}(t) \succeq \alpha \boldsymbol{I}. \tag{5.59}$$

Sei o. B. d. A. das (j, j)-te Element von $\boldsymbol{G}(t)$ gerade $\lambda_{\min}(\Xi_1(t))$ und $\boldsymbol{e}_j(t)$ der j-te Einheitsvektor. Dann folgt

$$\lambda_{\min,1}(t) = \lambda_{\min}(\Xi_1(t)) = \boldsymbol{e}_j^\mathsf{T}(t)\boldsymbol{G}(t)\boldsymbol{e}_j(t) \geq \alpha. \tag{5.60}$$

□

Lemma 5.11 (Eigenwertsignal $\lambda_{\min,2}(t)$ zur Überprüfung der PE-Eigenschaft)
Wenn $\forall t \geq t_0$ das Signal $\boldsymbol{\sigma}(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^h$ PE ist, so steigt $\forall t \geq t_0$ das Eigenwertsignal

$$\lambda_{\min,2}(t) := \lambda_{\min}(\Xi_2(t)) := \lambda_{\min}\left(\int_{t_0}^{t} \boldsymbol{\sigma}(\tau)\boldsymbol{\sigma}^\mathsf{T}(\tau)\,\mathrm{d}\tau\right) \tag{5.61}$$

monoton und es gilt $\lambda_{\min,2}(kT + t_0) \geq k\alpha$, $k \in \mathbb{N}_{\geq 0}$, $\alpha > 0$, $T > 0$.

Beweis:
Sei $\boldsymbol{\sigma}(t)$ PE. Unter Verwendung von Definition 5.4 folgt daher

$$\Xi_2(kT + t_0) = \sum_{j=0}^{k-1} \underbrace{\int_{jT+t_0}^{(j+1)T+t_0} \boldsymbol{\sigma}(\tau)\boldsymbol{\sigma}^\mathsf{T}(\tau)\,\mathrm{d}\tau}_{\succeq \alpha \boldsymbol{I}} \succeq k\alpha \boldsymbol{I}. \tag{5.62}$$

Analog zu (5.57)–(5.59) resultiert aus (5.62)

$$\lambda_{\min,2}(kT + t_0) = \lambda_{\min}(\Xi_2(kT + t_0)) \geq k\alpha. \tag{5.63}$$

Zudem steigt (5.61) monoton, da für $kT + t_0 < t < (k+1)T + t_0$

$$\Xi_2(t) = \Xi_2(kT + t_0) + \underbrace{\int_{kT+t_0}^{t} \boldsymbol{\sigma}(\tau)\boldsymbol{\sigma}^\intercal(\tau)\,\mathrm{d}\tau}_{\succeq \mathbf{0}} \succeq \Xi_2(kT) \tag{5.64}$$

gilt. □

Lemma 5.10 liefert mit dem Eigenwertsignal $\lambda_{\min,1}(t)$ und (5.55) ein notwendiges und hinreichendes Kriterium für die Erfüllung der PE-Eigenschaft. Für dessen Berechnung müsste jedoch ein festes T im Voraus bestimmt werden, was in der Simulation nicht praktikabel ist. Daher wird zunächst $\lambda_{\min,2}(t)$ betrachtet, um einen möglichen Kandidaten für T zu ermitteln. Vorteilhaft ist, dass das Signal $\lambda_{\min,2}(t)$ direkt zur Laufzeit berechnet werden kann. Es stellt nach Lemma 5.11 jedoch lediglich ein notwendiges Kriterium zur Erfüllung der PE-Eigenschaft dar. Nach Wahl eines numerischen Schwellwerts α_t wird der Zeitpunkt T_t bestimmt, zu dem erstmalig $\lambda_{\min,2} \geq \alpha_\mathrm{t}$ gilt. T_t wird dann als Kandidat zur Überprüfung der Ungleichung (5.55) mit $T = T_\mathrm{t}$ und $\alpha = \alpha_\mathrm{t}$ verwendet. Ist (5.55) erfüllt, so ist $\boldsymbol{\sigma}(t)$ nach Lemma 5.10 PE. Sollte hingegen $\lambda_{\min,1}(t)$ den Schwellwert α_t unterschreiten, so wird T_t iterativ erhöht. Bleibt auch hierbei $\lambda_{\min,1}(t)$ unterhalb von α_t, so wurde die Einhaltung der PE-Eigenschaft für $\boldsymbol{\sigma}(t)$ nicht nachgewiesen. Sollte das Eigenwertsignal $\lambda_{\min,2}(t)$ sättigendes Verhalten aufweisen, so beweist Lemma 5.11, dass $\boldsymbol{\sigma}(t)$ nicht PE ist.

5.6 Anregungssignale für ADP-basierte Differenzialspiele

Die bisher präsentierten Hauptbeiträge des vorliegenden Kapitels formulieren Bedingungen an den Systemzustand $\boldsymbol{x}$, um Konvergenz des Policy-Evaluation-Schrittes der Policy Iteration nach Algorithmus 5.1 zu gewährleisten. Während Satz 5.2 zeigt, dass hierbei eine genaue Analyse möglich ist, stellt die Frage nach geeigneten Stellgrößen $\boldsymbol{u}_i$, $i \in \mathcal{N}$, um diese Bedingungen an den Systemzustand $\boldsymbol{x}$ zu garantieren, insbesondere bei unbekannter Systemdynamik ein grundlegendes Problem dar. Auch für die Identifikation einer a priori unbekannten Systemdynamik wäre wiederum eine geeignete Anregung erforderlich (vgl. Abschnitt 2.3), sodass ein komplett analytischer Anregungssignalentwurf der fehlenden Systemkenntnis grundsätzlich widerspricht. Um jedoch zu Analysezwecken die Umsetzung von Simulationsbeispielen in Abschnitt 5.7 zu ermöglichen, wird die folgende Annahme verwendet[103].

Annahme 5.5
Für den Anregungssignalentwurf seien $\boldsymbol{f}(\boldsymbol{x})$, $\boldsymbol{g}(\boldsymbol{x})$, $\boldsymbol{\phi}_i(\boldsymbol{x})$, $\boldsymbol{R}_{ii}$ und $\hat{\boldsymbol{w}}_i^{[l]}$, $\forall i \in \mathcal{N}$, bekannt.

[103] Wenngleich die Spieler $i \in \mathcal{N}$ entsprechend Definition 5.1 weiterhin keine Kenntnis der Regelgesetze $\hat{\boldsymbol{\mu}}_j^{[l]}$, $j \in \mathcal{N}$, $j \neq i$, besitzen, wird allein für den Entwurf von $\boldsymbol{u}_\mathrm{ex}$ angenommen, dass $\hat{\boldsymbol{w}}_i^{[l]}$, $\forall i \in \mathcal{N}$, bekannt sind.

Im Folgenden wird ein additives Anregungssignal $\boldsymbol{u}_{\mathrm{ex}}$ gesucht, sodass die Critic-Gewichtsfehler $\tilde{\boldsymbol{w}}_i^{[l+1]} = \boldsymbol{w}_i^{[l+1]} - \hat{\boldsymbol{w}}_i^{[l+1]}$, $\forall i \in \mathcal{N}$, im Policy-Evaluation-Schritt (vgl. Algorithmus 5.1) exponentiell konvergieren. Hierzu ist nach Lemma 5.1 die Erfüllung der PE-Eigenschaft von $\boldsymbol{\sigma}_i$, $\forall i \in \mathcal{N}$, notwendig und hinreichend. Um eine Eingriffsmöglichkeit zur Systemanregung zu bieten, wird ein Anregungssignal $\boldsymbol{u}_{\mathrm{ex}}$ verwendet, das die im Folgenden formulierte Problemstellung lösen soll.

Problem 5.2

Sei Annahme 5.5 erfüllt. Gesucht ist ein Anregungssignal

$$\boldsymbol{u}_{\mathrm{ex}} = \begin{bmatrix} \boldsymbol{u}_{\mathrm{ex},1}^{\mathsf{T}} & \boldsymbol{u}_{\mathrm{ex},2}^{\mathsf{T}} & \dots & \boldsymbol{u}_{\mathrm{ex},N}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}, \tag{5.65}$$

sodass $\boldsymbol{\sigma}_i$, $\forall i \in \mathcal{N}$, PE ist. Dabei sind $\boldsymbol{u}_{\mathrm{ex},i}$ Anregungssignale, die nach Abbildung 5.1 auf die den N Spielern zur Verfügung stehenden Eingänge addiert werden, d. h. es gilt $\boldsymbol{u}_i = \hat{\boldsymbol{\mu}}_i^{[l]}(\boldsymbol{x}) + \boldsymbol{u}_{\mathrm{ex},i}$.

Wie in Abbildung 5.1 ersichtlich ist, wirkt jeder Spieler $i \in \mathcal{N}$ durch das jeweilige Regelgesetz $\hat{\boldsymbol{\mu}}_i^{[l]}(\boldsymbol{x})$ in der l-ten Iteration der Policy Iteration nach Algorithmus 5.1 auf das System ein. Basierend auf dem durch (5.15) gegebenen Gradientenabstieg werden die Critic-Gewichte $\hat{\boldsymbol{w}}_i^{[l+1]}$, $\forall i \in \mathcal{N}$, bis zur Konvergenz des Policy-Evaluation-Schrittes adaptiert. Anschließend

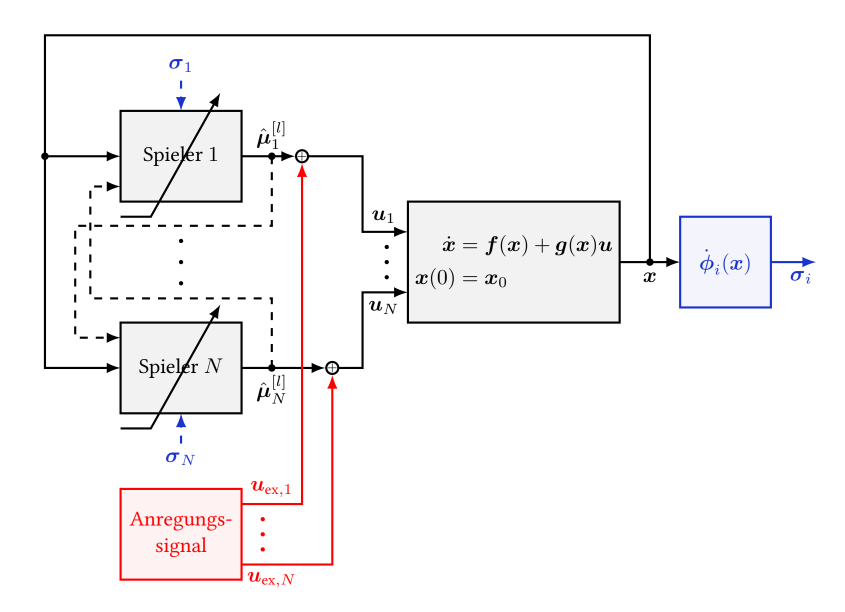


Abbildung 5.1: Regelkreisstruktur des eingangsaffinen Nicht-Nullsummen-Differenzialspiels mit Anregungssignal $\boldsymbol{u}_{\mathrm{ex}}$ (rot). Das Signal $\boldsymbol{\sigma}_i$ (blau) wird von jedem Spieler individuell berechnet und ist nicht als Systemgröße messbar, hier der Vollständigkeit halber jedoch dargestellt (Abbildung nach [Kar19]).

werden die Regelgesetze $\hat{\boldsymbol{\mu}}_i^{[l+1]}(\boldsymbol{x})$, $\forall i \in \mathcal{N}$, gemäß (5.24) angepasst. Die Anregung des Systems erfolgt hierbei nicht über zusätzliche Anregungssignale der einzelnen Spieler, sondern soll durch das externe Anregungssignal $\boldsymbol{u}_{\text{ex}}$ sichergestellt werden.

Die nachfolgende Bemerkung skizziert, weshalb häufig verwendetes weißes Rauschen als Anregungssignal $\boldsymbol{u}_{\text{ex}}$ (vgl. Abschnitt 2.3) aus theoretischer Sicht zwar plausibel, jedoch praktisch ungeeignet erscheint.

Bemerkung 5.4 (Weißes Rauschen als Anregungssignal $\boldsymbol{u}_{\text{ex}}$)
Nach Proposition 5.1 ist $\boldsymbol{\sigma} = \dot{\boldsymbol{\phi}}(\boldsymbol{x}_{\text{PE}})$ PE $\forall \boldsymbol{\omega} \in \Omega$, wenn es gelingt, eine Menge Ω zu bestimmen, für die $\forall \boldsymbol{\omega} \in \Omega$ und $\forall \boldsymbol{\alpha} \neq \mathbf{0}$ folgt, dass $\boldsymbol{\alpha}^\intercal \boldsymbol{\sigma}_i \neq 0$ gilt. Wird das Anregungssignal $\boldsymbol{u}_{\text{ex}}$ in (5.65) durch weißes Rauschen realisiert, so erscheint es für vollständig steuerbare Systeme und sinnvoll gestellte Probleme, d. h. unter der Annahme, dass eine ausreichende Anregung grundsätzlich möglich ist, unwahrscheinlich, dass sich alle Frequenzen durch die Transformation $\dot{\boldsymbol{\phi}}_i(\boldsymbol{x}_{\text{PE}})$ gegenseitig aufheben und $\boldsymbol{\alpha}^\intercal \boldsymbol{\sigma}_i = 0$ resultiert.
Jedoch sei zu erwähnen, dass eine Anregung technischer Systeme mit weißem Rauschen häufig nicht erwünscht ist. So sind beispielsweise hohe mechanische Belastungen zu erwarten, zudem dämpfen Systeme mit Tiefpassverhalten einen wesentlichen Teil der Anregungsenergie. Die Verwendung weißen Rauschens als Anregungssignal bietet also keine Freiheitsgrade, zusätzliche problem- oder systemspezifische Anforderungen zu berücksichtigen.

Im Folgenden werden die theoretischen Ergebnisse aus Abschnitt 5.4 und insbesondere Satz 5.2 verwendet, um ein Anregungssignal $\boldsymbol{u}_{\text{ex}}$ zu entwerfen und somit Simulationen zu ermöglichen. Der prinzipielle Ablauf zum Entwurf von $\boldsymbol{u}_{\text{ex}}$ ist in Abbildung 5.2 gezeigt. In einem ersten Schritt wird hierbei $\boldsymbol{x}_{\text{PE}}(t)$ in Übereinstimmung mit Annahme 5.4 gewählt. Die konkreten Frequenzen $\boldsymbol{\omega}$ sind zu diesem Zeitpunkt noch frei. Mithilfe von Satz 5.2 werden dann die Signale $\dot{\boldsymbol{\phi}}_i(\boldsymbol{x}_{\text{PE}})$, $\forall i \in \mathcal{N}$, analysiert und die Mengen Ω_i berechnet, sodass für jedes $\boldsymbol{\omega} \in \Omega_i$ folgt, dass $\boldsymbol{x}_{\text{PE}}$ SR bezüglich $\dot{\boldsymbol{\phi}}_i(\cdot)$ ist. Anschließend können verbliebene Freiheitsgrade gewählt werden. Einerseits beinhaltet dies die konkrete Wahl der Frequenzen in $\boldsymbol{x}_{\text{PE}}$ zu einem beliebigen $\boldsymbol{\omega}_{\text{c}} \in \Omega = \bigcap_i \Omega_i$, andererseits kann durch Wahl von $\boldsymbol{\nu} \in \mathbb{R}^n$, $\nu_j \neq 0$, eine Skalierung nach Lemma 5.9 erfolgen. Hierdurch können problemspezifische Anforderungen, wie beispielsweise die Tiefpasscharakteristik eines Systems, berücksichtigt werden. Schließlich wird im letzten Schritt ein Anregungssignal $\boldsymbol{u}_{\text{ex}}$ basierend auf

$$\bar{\boldsymbol{x}}_{\text{PE}} = \operatorname{diag}(\boldsymbol{\nu})\; \boldsymbol{x}_{\text{PE}}\big|_{\boldsymbol{\omega}=\boldsymbol{\omega}_{\text{c}}} \tag{5.66}$$

berechnet. Dieser letzte Schritt wird im Folgenden für das Beispiel eines exakt zustandslinearisierbaren Systems anhand eines flachheitsbasierten Vorsteuerungsansatzes betrachtet.

In einem ersten Schritt soll untersucht werden, ob alle $p = \sum_{i=1}^{N} p_i$ Systemeingänge zur Anregung benötigt werden. Zu diesem Zweck wird $\boldsymbol{g}(\boldsymbol{x})$ in (5.1) analysiert, um Spalten von

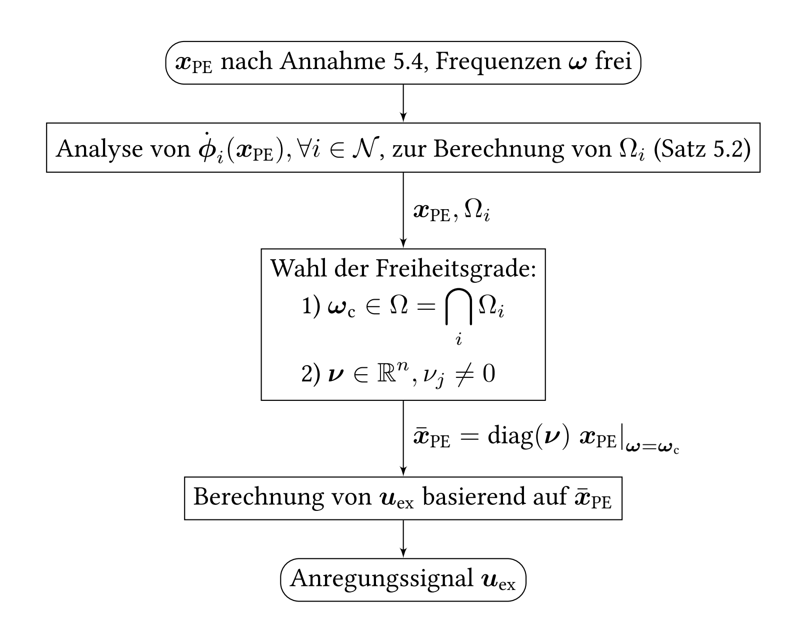


Abbildung 5.2: Ablaufdiagramm zum Design eines Anregungssignals $\boldsymbol{u}_{\mathrm{ex}}$.

$\boldsymbol{g}(\boldsymbol{x})$ zu identifizieren, die durch einen funktionalen Zusammenhang mittels anderer Spalten ausgedrückt werden können. Eine Umsortierung der Spalten von $\boldsymbol{g}(\boldsymbol{x})$ führt dann zu

$$\bar{\boldsymbol{g}}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{g}_{\mathrm{I}}(\boldsymbol{x}) & \boldsymbol{g}_{\mathrm{II}}(\boldsymbol{x}) \end{bmatrix}, \tag{5.67}$$

$\boldsymbol{g}_{\mathrm{I}} : \mathbb{R}^n \to \mathbb{R}^{n \times p_{\mathrm{I}}}$, $\boldsymbol{g}_{\mathrm{II}} : \mathbb{R}^n \to \mathbb{R}^{n \times p_{\mathrm{II}}}$ und der funktionalen Beziehung $\boldsymbol{\gamma}(\boldsymbol{x})$, sodass $\forall \boldsymbol{x} \in \mathcal{X}$ der Zusammenhang

$$\boldsymbol{g}_{\mathrm{II}}(\boldsymbol{x}) = \boldsymbol{g}_{\mathrm{I}}(\boldsymbol{x})\boldsymbol{\gamma}(\boldsymbol{x}) \tag{5.68}$$

besteht und $\boldsymbol{g}_{\mathrm{I}}(\boldsymbol{x})$ maximalen Spaltenrang aufweist, d. h. $\mathrm{Rang}\,(\boldsymbol{g}_{\mathrm{I}}(\boldsymbol{x})) = p_{\mathrm{I}}$ gilt. Werden die Systemeingänge in analoger Weise wie die Spalten von $\boldsymbol{g}(\boldsymbol{x})$ umsortiert, ergibt sich

$$\bar{\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{u}_{\mathrm{I}} \\ \boldsymbol{u}_{\mathrm{II}} \end{bmatrix} \tag{5.69}$$

mit $\bar{\boldsymbol{u}} \in \mathbb{R}^p$, $\boldsymbol{u}_{\mathrm{I}} \in \mathbb{R}^{p_{\mathrm{I}}}$ und $\boldsymbol{u}_{\mathrm{II}} \in \mathbb{R}^{p_{\mathrm{II}}}$. Da aufgrund von (5.68) eine Anregung des Systems über die zu $\boldsymbol{u}_{\mathrm{II}}$ zusammengefassten Eingänge keinen Mehrwert liefert, wird das Anregungssignal $\boldsymbol{u}_{\mathrm{ex,II}}$ der entsprechenden Eingänge zu null gesetzt. Für die Systemanregung werden lediglich die Eingänge $\boldsymbol{u}_{\mathrm{I}}$ verwendet, sodass $\boldsymbol{u}_{\mathrm{ex,I}}$ entworfen werden muss und sich das Anregungssignal $\boldsymbol{u}_{\mathrm{ex}}$ schließlich aus der Rücksortierung von $\bar{\boldsymbol{u}}_{\mathrm{ex}} = \begin{bmatrix} \boldsymbol{u}_{\mathrm{ex,I}}^{\mathsf{T}} & \boldsymbol{u}_{\mathrm{ex,II}}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} \boldsymbol{u}_{\mathrm{ex,I}}^{\mathsf{T}} & \boldsymbol{0}_{1 \times p_{\mathrm{II}}} \end{bmatrix}^{\mathsf{T}}$ ergibt[104].

[104] Dieses Vorgehen, um die Anzahl der Eingänge zur Systemanregung zu reduzieren, ist generell anwendbar und lässt sich somit beispielsweise auch bei klassischer Anregung durch weißes Rauschen nutzen.

Das globale Anregungssignal $\boldsymbol{u}_{\text{ex}}$ (bzw. $\boldsymbol{u}_{\text{ex,I}}$) kann zu Simulationszwecken mithilfe eines inversionsbasierten Vorsteuerungsansatzes (siehe beispielsweise [DCP96], [Hir79], [Dev02]) entworfen werden. Im Folgenden wird für das Beispiel eines (global) exakt zustandslinearisierbaren Systems (vgl. [Ada18, Definition 29]), das folglich auch differenziell flach[105] ist [Ada18, Satz 67], eine flachheitsbasierte Vorsteuerung vorgestellt und analysiert. Dazu werde zunächst allgemein

$$\dot{\boldsymbol{x}} = \boldsymbol{f}_{\text{g}}(\boldsymbol{x}) + \boldsymbol{g}_{\text{I}}(\boldsymbol{x})\boldsymbol{u}_{\text{I}} \tag{5.70}$$

mit

$$\boldsymbol{f}_{\text{g}}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\hat{\boldsymbol{\mu}}^{[l]}(\boldsymbol{x}) \tag{5.71}$$

betrachtet. Somit entspricht $\boldsymbol{f}_{\text{g}}$ der Dynamik des durch $\hat{\boldsymbol{\mu}}^{[l]}(\boldsymbol{x})$ geschlossenen Regelkreises des Systems (5.1). Dieses System lässt sich auf Byrnes-Isidori-Normalform (BINF) [Ada18, S. 354] ohne interne Dynamik transformieren, falls eine (ggf. fiktive) Ausgangsfunktion $\boldsymbol{h}(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^{p_{\text{I}}}$ mit vollem totalen relativen Grad[106] [Ada18, Definition 28] ($\delta = \sum_j \delta_j = n$) existiert [Zim84]. Nach [Isi89, S. 245] ist das System in BINF dann durch p_{I} Subsysteme der Form

$$\begin{aligned}
\dot{z}_{\text{B},1,j} &= z_{\text{B},2,j}, \\
&\vdots \\
\dot{z}_{\text{B},\delta_j-1,j} &= z_{\text{B},\delta_j,j}, \\
\dot{z}_{\text{B},\delta_j,j} &= \underbrace{L_{\boldsymbol{f}_{\text{g}}}^{\delta_j} h_j(\boldsymbol{x})}_{\beta_j(\boldsymbol{x})} + \underbrace{\begin{bmatrix} L_{\boldsymbol{g}_{\text{I},1}} L_{\boldsymbol{f}_{\text{g}}}^{\delta_j-1} h_j(\boldsymbol{x}) & \dots & L_{\boldsymbol{g}_{\text{I},p_{\text{I}}}} L_{\boldsymbol{f}_{\text{g}}}^{\delta_j-1} h_j(\boldsymbol{x}) \end{bmatrix}}_{\boldsymbol{d}_j^{\mathsf{T}}(\boldsymbol{x})} \boldsymbol{u}_{\text{I}},
\end{aligned} \tag{5.72}$$

$j \in \{1, \dots, p_{\text{I}}\}$, gegeben. Dabei bezeichnet $h_j(\boldsymbol{x})$ das j-te Element von $\boldsymbol{h}(\boldsymbol{x})$, L die Lie-Ableitung[107], δ_j den relativen Grad von (5.70) bezüglich $h_j(\boldsymbol{x})$ und $\boldsymbol{g}_{\text{I},k}(\boldsymbol{x})$, $k \in \{1, \dots, p_{\text{I}}\}$, die k-te Spalte von $\boldsymbol{g}_{\text{I}}(\boldsymbol{x})$. Der Diffeomorphismus

$$\begin{aligned}
\boldsymbol{z}_{\text{B}} &= \boldsymbol{t}(\boldsymbol{x}) \\
&= \begin{bmatrix} t_{1,1}(\boldsymbol{x}) & \cdots & t_{\delta_1,1}(\boldsymbol{x}) & t_{1,2}(\boldsymbol{x}) & \cdots & t_{\delta_2,2}(\boldsymbol{x}) & t_{1,p_{\text{I}}}(\boldsymbol{x}) & \cdots & t_{\delta_{p_{\text{I}}},p_{\text{I}}}(\boldsymbol{x}) \end{bmatrix}^{\mathsf{T}},
\end{aligned} \tag{5.73}$$

---

105 Ein System $\dot{\boldsymbol{x}} = \boldsymbol{f}_{\text{dyn}}(\boldsymbol{x}, \boldsymbol{u})$ mit Rang $\left(\frac{\partial \boldsymbol{f}_{\text{dyn}}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}\right) = \dim(\boldsymbol{u})$ heißt flach (vgl. [FLMR95], [Ada18, Kapitel 3.2]), wenn ein (realer oder fiktiver) flacher Ausgang $\boldsymbol{y} = \boldsymbol{h}_\Gamma\left(\boldsymbol{x}, \boldsymbol{u}, \dot{\boldsymbol{u}}, \dots, \boldsymbol{u}^{(\chi_1)}\right)$, $\chi_1 \in \mathbb{N}$, χ_1 endlich, $\dim(y) = \dim(u)$, existiert, sodass $\exists \chi_2 \in \mathbb{N}$, χ_2 endlich, mit $\boldsymbol{x} = \boldsymbol{\Gamma}_1\left(\boldsymbol{y}, \dot{\boldsymbol{y}}, \dots, \boldsymbol{y}^{(\chi_2)}\right)$ und $\boldsymbol{u} = \boldsymbol{\Gamma}_2\left(\boldsymbol{y}, \dot{\boldsymbol{y}}, \dots, \boldsymbol{y}^{(\chi_2+1)}\right)$ [Ada18, Definition 27], [Lév09, Definition 6.1].

106 Der totale relative Grad δ eines Systems ist auch als Differenzordnung eines Systems bekannt. Eine solche Ausgangsfunktion lässt sich bei einfachen Systemen häufig durch Betrachtung der Systemmatrizen ermitteln, ansonsten kann im Fall der betrachteten eingangsaffinen Systeme die in [Zim84] beschriebene Methode verwendet werden.

107 $L_{\boldsymbol{f}}^j h(\boldsymbol{x}) := L_{\boldsymbol{f}} L_{\boldsymbol{f}}^{j-1} h(\boldsymbol{x})$ bezeichnet die j-te Lie-Ableitung von $h(\boldsymbol{x})$ bezüglich $\boldsymbol{f}(\boldsymbol{x})$. Dabei gilt per Definition $L_{\boldsymbol{f}} h(\boldsymbol{x}) := \frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})$.

der den Systemzustand $\boldsymbol{x}$ auf BINF transformiert, ergibt sich zu

$$\begin{aligned} z_{\mathrm{B},1,j} &= t_{1,j}(\boldsymbol{x}) = h_j(\boldsymbol{x}), \\ z_{\mathrm{B},2,j} &= t_{2,j}(\boldsymbol{x}) = L_{\boldsymbol{f}_\mathrm{g}} h_j(\boldsymbol{x}), \\ &\vdots \\ z_{\mathrm{B},\delta_j,j} &= t_{\delta_j,j}(\boldsymbol{x}) = L_{\boldsymbol{f}_\mathrm{g}}^{\delta_j-1} h_j(\boldsymbol{x}). \end{aligned} \tag{5.74}$$

Werden nun die Elemente $z_{\mathrm{B},1,j}, \forall j \in \{1,\ldots,p_\mathrm{I}\}$, verwendet, um

$$\boldsymbol{y}_\mathrm{f} = \boldsymbol{h}_\mathrm{f}(\boldsymbol{z}_\mathrm{B}) = \begin{bmatrix} z_{\mathrm{B},1,1} & z_{\mathrm{B},1,2} & \ldots & z_{\mathrm{B},1,p_\mathrm{I}} \end{bmatrix}^\mathsf{T} \tag{5.75}$$

als Kandidat für einen flachen Ausgang zu wählen, so folgt

$$\begin{aligned} \boldsymbol{z}_\mathrm{B} &= \begin{bmatrix} z_{\mathrm{B},1,1} & \dot{z}_{\mathrm{B},1,1} & \ldots & z_{\mathrm{B},1,1}^{(\delta_1-1)} & \ldots & z_{\mathrm{B},1,p_\mathrm{I}} & \dot{z}_{\mathrm{B},1,p_\mathrm{I}} & \ldots & z_{\mathrm{B},1,p_\mathrm{I}}^{(\delta_{p_\mathrm{I}}-1)} \end{bmatrix}^\mathsf{T} \\ &=: \boldsymbol{\Psi}_1 \left(\boldsymbol{y}_\mathrm{f}, \dot{\boldsymbol{y}}_\mathrm{f}, \ldots, \boldsymbol{y}_\mathrm{f}^{(\delta_\mathrm{max}-1)} \right) \end{aligned} \tag{5.76}$$

mit $\delta_\mathrm{max} = \max\{\delta_1,\ldots,\delta_{p_\mathrm{I}}\}$. Aus (5.72) folgt

$$\begin{aligned} \boldsymbol{u}_\mathrm{I} &= \boldsymbol{D}^{-1}(\boldsymbol{x}) \left(-\boldsymbol{\beta}(\boldsymbol{x}) + \begin{bmatrix} \dot{z}_{\mathrm{B},\delta_1,1} \\ \vdots \\ \dot{z}_{\mathrm{B},\delta_{p_\mathrm{I}},p_\mathrm{I}} \end{bmatrix} \right) = \boldsymbol{D}^{-1}(\boldsymbol{x}) \left(-\boldsymbol{\beta}(\boldsymbol{x}) + \begin{bmatrix} y_{\mathrm{f},1}^{(\delta_1)} \\ \vdots \\ y_{\mathrm{f},p_\mathrm{I}}^{(\delta_{p_\mathrm{I}})} \end{bmatrix} \right) \\ &= \boldsymbol{\Psi}_2 \left(\boldsymbol{y}_\mathrm{f}, \dot{\boldsymbol{y}}_\mathrm{f}, \ldots, \boldsymbol{y}_\mathrm{f}^{(\delta_\mathrm{max})} \right), \end{aligned} \tag{5.77}$$

wobei die Vektoren $\boldsymbol{d}_j^\mathsf{T}(\boldsymbol{x})$ zur sogenannten Entkopplungsmatrix $\boldsymbol{D}(\boldsymbol{x})$ [Ada18, S. 373] zusammengefasst werden, sich der Vektor $\boldsymbol{\beta}(\boldsymbol{x})$ aus $\beta_j(\boldsymbol{x})$ zusammensetzt und im letzten Schritt $\boldsymbol{x} = \boldsymbol{t}^{-1}(\boldsymbol{\Psi}_1(\cdot))$ genutzt wird. Falls nun $\det(\boldsymbol{D}(\boldsymbol{x})) \neq 0$ gilt, so existiert $\boldsymbol{\Psi}_2$ in (5.77) und der gewählte Kandidat $\boldsymbol{h}_\mathrm{f}(\boldsymbol{z}_\mathrm{B})$ stellt tatsächlich einen flachen Ausgang dar, da $\boldsymbol{\Psi}_1$ und $\boldsymbol{\Psi}_2$ existieren. Da $\boldsymbol{\Psi}_2$ nach [FLMR95] die Systemdynamik invertiert, kann direkt eine Vorsteuerung entworfen werden. Hierzu wird für die hier betrachtete Problemstellung die gewählte Anregungstrajektorie $\bar{\boldsymbol{x}}_\mathrm{PE}$ mit dem Diffeomorphismus $\boldsymbol{t}(\cdot)$ in BINF überführt und mithilfe von $\boldsymbol{y}_\mathrm{f}|_{\bar{\boldsymbol{x}}_\mathrm{PE}} = \boldsymbol{h}_\mathrm{f}\left(\boldsymbol{z}_\mathrm{B}|_{\bar{\boldsymbol{x}}_\mathrm{PE}}\right) = \boldsymbol{h}_\mathrm{f}(\boldsymbol{t}(\bar{\boldsymbol{x}}_\mathrm{PE}))$ das Anregungssignal $\boldsymbol{u}_\mathrm{ex,I}$ aus (5.77) berechnet. Schließlich resultiert das Anregungssignal $\boldsymbol{u}_\mathrm{ex}$ aus der Rücksortierung von $\bar{\boldsymbol{u}}_\mathrm{ex} = \begin{bmatrix} \boldsymbol{u}_\mathrm{ex,I}^\mathsf{T} & \boldsymbol{u}_\mathrm{ex,II}^\mathsf{T} \end{bmatrix}^\mathsf{T}$, wobei $\boldsymbol{u}_\mathrm{ex,II} = \boldsymbol{0}$ gilt.

Für die nachfolgenden Analysen bezeichne

$$\check{\boldsymbol{x}} = \boldsymbol{t}^{-1} \left(\boldsymbol{\Psi}_1 \left(\boldsymbol{y}_\mathrm{f}|_{\bar{\boldsymbol{x}}_\mathrm{PE}}, \dot{\boldsymbol{y}}_\mathrm{f}|_{\bar{\boldsymbol{x}}_\mathrm{PE}}, \ldots, \boldsymbol{y}_\mathrm{f}^{(\delta_\mathrm{max}-1)}\Big|_{\bar{\boldsymbol{x}}_\mathrm{PE}} \right) \right) \tag{5.78}$$

den idealen Systemzustand, der mit exakter Systeminversion und ohne Anfangsfehler des Zustands, d. h. ohne Einschwingvorgänge, resultieren würde. Die Definition der Flachheit [Ada18, Definition 27] stellt insbesondere durch $\dim(\boldsymbol{y}_\mathrm{f}) = \dim(\boldsymbol{u}_\mathrm{I})$ sicher, dass die Ausgänge

$y_{\mathrm{f},i}, i \in \{1, \dots, p_{\mathrm{I}}\}$, differenziell unabhängig sind. Dies ermöglicht, dass die Wahl der (fiktiven) Ausgangsgröße $\boldsymbol{y}_{\mathrm{f}}$ beliebig ist und durch die Vorsteuerung auch erzielt werden kann (vgl. [Ada18, S. 210 f.]). Falls jedoch nicht der gesamte Zustandsvektor $\boldsymbol{x}$ den flachen Ausgang $\boldsymbol{y}_{\mathrm{f}}$ darstellt, so können im Allgemeinen die Elemente des Zustandsvektors nicht unabhängig voneinander vorgegeben werden, sondern können differenzielle Abhängigkeiten aufweisen. In diesem Fall kann $\check{\boldsymbol{x}} = \boldsymbol{x}_{\mathrm{PE}}$ im Allgemeinen nicht erreicht werden.

Im Folgenden wird daher formal untersucht, unter welchen Annahmen trotz einer Abweichung des idealen Zustands $\check{\boldsymbol{x}}$ von $\bar{\boldsymbol{x}}_{\mathrm{PE}}$, beispielsweise durch differenzielle Abhängigkeiten, sowie möglichen Abweichungen, die der tatsächliche Systemzustand $\boldsymbol{x}$ von $\check{\boldsymbol{x}}$ haben kann[108], die Erfüllung der PE-Eigenschaft von $\boldsymbol{\sigma}_i(t)\ \forall i \in \mathcal{N}$ gewährleistet werden kann. Hierzu gelte die folgende Annahme.

Annahme 5.6

Für $\check{\boldsymbol{x}}$ sei

$$\check{\boldsymbol{\sigma}}_i := \boldsymbol{\sigma}_i(\check{\boldsymbol{x}}(t)) = \dot{\boldsymbol{\phi}}_i(\check{\boldsymbol{x}}(t)) = \boldsymbol{M}_i \frac{\mathrm{d}\boldsymbol{v}_{\mathrm{freq}}(t)}{\mathrm{d}t} \tag{5.79}$$

mit $\boldsymbol{M}_i \in \mathbb{R}^{h_i \times h_{\mathrm{freq}}}$, $h_i \leq h_{\mathrm{freq}}$, und $\boldsymbol{v}_{\mathrm{freq}}(t) \in \mathbb{R}^{h_{\mathrm{freq}}}$, wobei $\boldsymbol{v}_{\mathrm{freq}}(t)$ ein Vektor ist, dessen Elemente Sinus- und Kosinusfunktionen unterschiedlicher Frequenzen (d. h. unterschiedlich für identische trigonometrische Funktionen) darstellen.
Der tatsächliche Systemzustand $\boldsymbol{x}$, der unter Verwendung der Regelgesetze $\hat{\boldsymbol{\mu}}_1^{[l]}(\boldsymbol{x}), \dots, \hat{\boldsymbol{\mu}}_N^{[l]}(\boldsymbol{x})$ und des Anregungssignals $\boldsymbol{u}_{\mathrm{ex}}$ resultiert, führe zu

$$\boldsymbol{\sigma}_i(t) = \check{\boldsymbol{\sigma}}_i(t) + \boldsymbol{\varepsilon}_{1,i}(t) + \boldsymbol{\varepsilon}_{2,i}(t), \qquad \forall i \in \mathcal{N}, \tag{5.80}$$

mit stetigen, beschränkten Signalen $\boldsymbol{\varepsilon}_{1,i}(t), \boldsymbol{\varepsilon}_{2,i}(t) : \mathbb{R}_{\geq 0} \to \mathbb{R}^{h_i}$. Zudem gelte $\|\boldsymbol{\varepsilon}_{1,i}(t)\|_2 \leq \bar{\varepsilon}_{\sigma,i}$, $\forall t$, und $\boldsymbol{\varepsilon}_{2,i}(t) \to \boldsymbol{0}$ für $t \to \infty$.

Bemerkung 5.5

Aufgrund von (5.79) muss für die nachfolgenden Aussagen somit nicht $\check{\boldsymbol{x}} = \bar{\boldsymbol{x}}_{\mathrm{PE}}$ gelten, sondern differenzielle Abhängigkeiten können berücksichtigt werden. Ob (5.79) gilt, lässt sich anhand von $\boldsymbol{\Psi}_1$, dem Diffeomorphismus $\boldsymbol{t}(\cdot)$ zur Transformation auf BINF und dem flachen Ausgang $\boldsymbol{y}_{\mathrm{f}}$ a priori überprüfen. Zudem entsprechen die in $\boldsymbol{v}_{\mathrm{freq}}$ auftretenden Frequenzen Linearkombinationen der Einträge von $\boldsymbol{\omega}_{\mathrm{c}}$. Weiterhin berücksichtigt (5.80) durch $\boldsymbol{\varepsilon}_{1,i}(t)$ Modellungenauigkeiten, beispielsweise der Regelgesetze oder der Systemdynamik, und durch $\boldsymbol{\varepsilon}_{2,i}(t)$ transiente Vorgänge. Beides kann zu Abweichungen zwischen $\boldsymbol{x}$ und $\check{\boldsymbol{x}}$ führen.

Das nachfolgende Lemma liefert eine theoretische Aussage hinsichtlich der Erfüllung der PE-Bedingung für das aus dem tatsächlichen Zustand $\boldsymbol{x}(t)$ resultierende Signal $\boldsymbol{\sigma}_i(\boldsymbol{x}(t))$.

108 Insbesondere transiente Vorgänge während des Einschwingvorgangs können zu $\boldsymbol{x} \neq \check{\boldsymbol{x}}$ führen.

Lemma 5.12

Sei $\bar{\boldsymbol{x}}_{\mathrm{PE}}(t) = \mathrm{diag}(\boldsymbol{\nu})\, \boldsymbol{x}_{\mathrm{PE}}(t)|_{\boldsymbol{\omega}=\boldsymbol{\omega}_{\mathrm{c}}}$ *wie in Lemma 5.9 mit entsprechender Wahl von* $\boldsymbol{\omega}_{\mathrm{c}} \in \Omega$ *und* $\boldsymbol{\nu}$ *gegeben. Unter Annahme 5.6 ist* $\boldsymbol{\sigma}_i(t) = \boldsymbol{\sigma}_i(\boldsymbol{x}(t))\ \forall i \in \mathcal{N}$ *PE, falls* $\bar{\varepsilon}_{\sigma,i}$ *hinreichend klein ist und* $\mathrm{Rang}(\boldsymbol{M}_i) = h_i$ *gilt.*

Beweis:
Der Beweis erfolgt in zwei Schritten. Im ersten Schritt wird gezeigt, dass $\boldsymbol{\sigma}_i(t)$ PE ist, falls $\check{\boldsymbol{\sigma}}_i(t)$ die PE-Eigenschaft erfüllt. Im zweiten Schritt wird bewiesen, dass $\check{\boldsymbol{\sigma}}_i(t)$ PE ist. Mit (5.80) folgt aus (5.19)

$$\begin{aligned}
&\frac{1}{T}\int_t^{t+T} |(\check{\boldsymbol{\sigma}}_i(\tau) + \boldsymbol{\varepsilon}_{1,i}(\tau) + \boldsymbol{\varepsilon}_{2,i}(\tau))^{\mathsf{T}}\, \boldsymbol{e}|\, \mathrm{d}\tau \\
&\geq \frac{1}{T}\int_t^{t+T} |(\check{\boldsymbol{\sigma}}_i(\tau))^{\mathsf{T}}\, \boldsymbol{e}|\, \mathrm{d}\tau - \frac{1}{T}\int_t^{t+T} |(\boldsymbol{\varepsilon}_{1,i}(\tau))^{\mathsf{T}}\, \boldsymbol{e}|\, \mathrm{d}\tau - \frac{1}{T}\int_t^{t+T} |(\boldsymbol{\varepsilon}_{2,i}(\tau))^{\mathsf{T}}\, \boldsymbol{e}|\, \mathrm{d}\tau \\
&\geq \alpha_{\mathrm{I},i} - \bar{\varepsilon}_{\sigma,i} - \frac{1}{T} M_{\varepsilon}(T). \qquad (5.81)
\end{aligned}$$

Die letzte Ungleichheit in (5.81) ergibt sich, wenn $\check{\boldsymbol{\sigma}}_i$ PE ist und aufgrund der Beschränktheit von $\boldsymbol{\varepsilon}_{1,i}$ sowie dem Konvergenzverhalten von $\boldsymbol{\varepsilon}_{2,i}$. Letzteres stellt

$$\int_t^{t+T} |(\boldsymbol{\varepsilon}_{2,i}(\tau))^{\mathsf{T}}\, \boldsymbol{e}|\, \mathrm{d}\tau \leq M_{\varepsilon}(T) \qquad (5.82)$$

sicher, wobei $M_{\varepsilon}(T)$ eine von T abhängige obere Schranke darstellt. Da $M_{\varepsilon}(T)$ aufgrund von $\boldsymbol{\varepsilon}_{2,i}(t) \to \mathbf{0}$ $(t \to \infty)$ mit steigendem T in die Sättigung übergeht, existiert $\bar{T}$, sodass $\alpha_{\mathrm{I},i} > \bar{\varepsilon}_{\sigma,i} + \frac{1}{T} M_{\varepsilon}(T), \forall T > \bar{T}$, gilt, wenn $\bar{\varepsilon}_{\sigma,i}$ hinreichend klein ist. Nach (5.19) folgt, dass $\boldsymbol{\sigma}_i$ ein PE-Signal ist, wenn $\check{\boldsymbol{\sigma}}_i$ PE ist. Dass $\check{\boldsymbol{\sigma}}_i$ ein PE-Signal ist, folgt schlussendlich aus (5.79) unter Anwendung von [NA05, Lemma 6.1], da $\boldsymbol{v}_{\mathrm{freq}}(t)$ PE ist und $\boldsymbol{M}_i$ maximalen Zeilenrang besitzt. Die PE-Eigenschaft von $\boldsymbol{v}_{\mathrm{freq}}(t)$ ist dabei ein direktes Resultat aus Lemma 5.3. □

In Abschnitt 5.4 wurden zentrale theoretische Zusammenhänge zwischen dem Systemzustand $\boldsymbol{x}$ und der Erfüllung der PE-Eigenschaft von $\boldsymbol{\sigma}_i$ analysiert und somit Problem 5.1 gelöst. Darauf aufbauend wurde in Abschnitt 5.6, ausgehend von Problem 5.2, ein mögliches Entwurfsverfahren für Anregungssignale ADP-basierter Differenzialspiele vorgestellt, welches für die Umsetzung der nachfolgenden Simulationsbeispiele genutzt wird.

5.7 Simulationsbeispiel zur Anregung von ADP-basierten Differenzialspielen

In diesem Abschnitt wird zunächst ein Beispiel eines eingangsaffinen Nicht-Nullsummen-Differenzialspiels vorgestellt. Anschließend werden gemäß dem in Abbildung 5.2 gezeigten

Ablauf Anregungssignale konstruiert. Diese Anregungssignale sowie ein Vergleichssignal, das weißes Rauschen approximiert, werden anschließend zur Anregung eines Beispielsystems verwendet. Die Adaption der Critic-Gewichte wird hierfür in der Simulation betrachtet.

5.7.1 Beispielproblem

Um die Konvergenz der Critic-Gewichte anhand von Simulationen zu untersuchen, wird zunächst ein beispielhaftes Differenzialspiel definiert. Damit die durch $\boldsymbol{w}_i^*, \forall i \in \mathcal{N}$, beschriebene Lösung zu Vergleichszwecken bekannt ist, wird ein Beispielsystem mit $N = 2$ Spielern, das mithilfe des Converse-HJB-Ansatzes [NP96] konstruiert wird, verwendet[109]. Nach Wahl der optimalen Value Functions zu

$$V_1^*(\boldsymbol{x}) := \frac{1}{2}x_1^2 + x_2^2, \qquad V_2^*(\boldsymbol{x}) := \frac{1}{4}x_1^2 + \frac{1}{2}x_2^2 \tag{5.83}$$

und der Gütefunktionale der beiden Spieler zu

$$\begin{aligned} J_1(\boldsymbol{x}_0, \mu_1, \mu_2) &:= \int_0^\infty \underbrace{2(x_1^2 + x_2^2)}_{=:q_1(\boldsymbol{x})} + \underbrace{2\left(\mu_1(\boldsymbol{x})\right)^2}_{=:\boldsymbol{\mu}_1^\mathsf{T}\boldsymbol{R}_{11}\boldsymbol{\mu}_1} + \underbrace{2\left(\mu_2(\boldsymbol{x})\right)^2}_{=:\boldsymbol{\mu}_2^\mathsf{T}\boldsymbol{R}_{12}\boldsymbol{\mu}_2} \,\mathrm{d}\tau, \\ J_2(\boldsymbol{x}_0, \mu_1, \mu_2) &:= \int_0^\infty \underbrace{x_1^2 + x_2^2}_{=:q_2(\boldsymbol{x})} + \underbrace{\left(\mu_1(\boldsymbol{x})\right)^2}_{=:\boldsymbol{\mu}_1^\mathsf{T}\boldsymbol{R}_{21}\boldsymbol{\mu}_1} + \underbrace{\left(\mu_2(\boldsymbol{x})\right)^2}_{=:\boldsymbol{\mu}_2^\mathsf{T}\boldsymbol{R}_{22}\boldsymbol{\mu}_2} \,\mathrm{d}\tau \end{aligned} \tag{5.84}$$

(vgl. (5.2)) liefert der Converse-HJB-Ansatz das nichtlineare, eingangsaffine System

$$\begin{aligned} \dot{\boldsymbol{x}} = &\underbrace{\begin{bmatrix} -2x_1 + x_2 \\ -x_2 - \frac{1}{2}x_1 + \frac{1}{4}x_2\left((\cos(2x_1)+2)^2 + (\sin\left(4x_1^2\right)+2)^2\right) \end{bmatrix}}_{=:\boldsymbol{f}(\boldsymbol{x})} \\ &+ \underbrace{\begin{bmatrix} 0 \\ \cos(2x_1)+2 \end{bmatrix}}_{=:\boldsymbol{g}_1(\boldsymbol{x})} u_1 + \underbrace{\begin{bmatrix} 0 \\ \sin\left(4x_1^2\right)+2 \end{bmatrix}}_{=:\boldsymbol{g}_2(\boldsymbol{x})} u_2 \end{aligned} \tag{5.85}$$

(vgl. (5.1)) als mögliche Lösung[110]. Der initiale Systemzustand wird zu $\boldsymbol{x}_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^\mathsf{T}$ gesetzt[111]. Die Basisfunktionen

$$\boldsymbol{\phi}_1(\boldsymbol{x}) = \boldsymbol{\phi}_2(\boldsymbol{x}) := \begin{bmatrix} x_1^2 & x_1x_2 & x_2^2 \end{bmatrix}^\mathsf{T}, \tag{5.86}$$

[109] Ähnliche Testsysteme werden auch in der Literatur konstruiert, siehe beispielsweise [VL11], [LLW14].

[110] Das sich im Zuge des Converse-HJB-Ansatzes ergebende Gleichungssystem ist unterbestimmt, da den beiden skalaren HJB-Gleichungen der Spieler mit den Funktionen $\boldsymbol{f}(\boldsymbol{x})$, $\boldsymbol{g}_1(\boldsymbol{x})$ und $\boldsymbol{g}_2(\boldsymbol{x})$ (jeweils $\in \mathbb{R}^2$) insgesamt sechs unbekannte Größen gegenüberstehen.

[111] Eine initiale Auslenkung des Systemzustands würde eine zusätzliche Anregung des Systems bedeuten und wird daher zugunsten der Vergleichbarkeit der nachfolgend untersuchten Anregungssignale vermieden. Generell muss jedoch nicht $\boldsymbol{x}_0 = \boldsymbol{0}$ gelten.

die Annahme 5.3 erfüllen, können die optimalen Value Functions (5.83) parametrieren und die als unbekannt angenommenen gesuchten Nash-Gewichte ergeben sich zu

$$\boldsymbol{w}_1^* = \begin{bmatrix} \frac{1}{2} \\ 0 \\ 1 \end{bmatrix}, \qquad \boldsymbol{w}_2^* = \begin{bmatrix} \frac{1}{4} \\ 0 \\ \frac{1}{2} \end{bmatrix}. \tag{5.87}$$

5.7.2 Konstruktion geeigneter Anregungssignale

Um, wie in Abschnitt 5.6 beschrieben und in Abbildung 5.2 veranschaulicht, ein Anregungssignal $\boldsymbol{u}_{\text{ex}}$ zur Lösung von Problem 5.2 zu konstruieren, wird zunächst $\boldsymbol{x}_{\text{PE}}$ nach Annahme 5.4 gewählt. Anschließend wird die Menge Ω an Frequenzbedingungen nach Satz 5.2 berechnet. Nach Wahl der verbliebenen Freiheitsgrade $\boldsymbol{\omega}_{\text{c}} \in \Omega$ und $\boldsymbol{\nu} \in \mathbb{R}^n, \nu_j \neq 0$, wird schließlich $\boldsymbol{u}_{\text{ex}}$ mithilfe eines flachheitsbasierten Vorsteuerungsentwurfs berechnet.

5.7.2.1 Wahl geeigneter Anregungstrajektorien $\boldsymbol{x}_{\text{PE}}$

In Übereinstimmung mit Annahme 5.4 wird

$$\boldsymbol{x}_{\text{PE}} = \begin{bmatrix} \sin(\omega_1 t) + \sin(\omega_2 t) \\ \sin(\omega_3 t) \end{bmatrix} \tag{5.88}$$

gewählt, weshalb im Folgenden Bedingungen an $\boldsymbol{\omega} = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^\mathsf{T}$ untersucht werden. Die sich ergebenden Frequenzbedingungen Ω nach Satz 5.2 sind, wie in Lemma 5.6 gezeigt, durch Z Matrizen $\boldsymbol{C}_z$ definiert und durch

$$\Omega = \left\{ \boldsymbol{\omega} : \bigwedge_{z=1}^{Z} \boldsymbol{C}_z \boldsymbol{\omega} \neq \boldsymbol{0} \right\}, \quad Z \in \mathbb{N}_{\geq 1}, \tag{5.89}$$

gegeben. Die Matrizen $\boldsymbol{C}_z$ sind in Tabelle 5.1 aufgelistet[112]. Anschaulich betrachtet ist jeder Frequenzvektor $\boldsymbol{\omega}$ geeignet, der keine der $Z = 21$ Gleichungen $\boldsymbol{C}_z\boldsymbol{\omega} = \boldsymbol{0}$, $z = 1, \ldots, Z$, erfüllt. Durch die in Tabelle 5.1 gegebenen Matrizen $\boldsymbol{C}_z$ werden 21 Ebenen $\boldsymbol{C}_z\boldsymbol{\omega} = \boldsymbol{0}$ im $\mathbb{R}^3$ beschrieben, die alle durch den Koordinatenursprung verlaufen und auf denen der Frequenzvektor $\boldsymbol{\omega}$ nach (5.89) nicht liegen darf. In Abbildung 5.3 sind exemplarisch sechs dieser Ebenen veranschaulicht. Die Menge Ω, für die nach Satz 5.2 folgt, dass $\boldsymbol{\sigma} = \dot{\boldsymbol{\phi}}(\boldsymbol{x}_{\text{PE}})\ \forall \boldsymbol{\omega} \in \Omega$ PE ist, wird durch den Raum $\mathbb{R}^3$ ohne diese 21 Ebenen beschrieben.

In Übereinstimmung mit Lemma 5.7 existieren somit geeignete Frequenzen $\omega \in \Omega$, da $\Omega \neq \emptyset$ gilt. Durch die Wahl der verbliebenen Freiheitsgrade $\boldsymbol{\omega}_{\text{c}} \in \Omega$ und $\boldsymbol{\nu} \in \mathbb{R}^n, \nu_j \neq 0$, nach Lemma 5.9 folgt somit

$$\bar{\boldsymbol{x}}_{\text{PE}}(t) = \begin{bmatrix} \nu_1 & 0 \\ 0 & \nu_2 \end{bmatrix} \begin{bmatrix} \sin(\omega_1 t) + \sin(\omega_2 t) \\ \sin(\omega_3 t) \end{bmatrix}. \tag{5.90}$$

[112] Die Berechnung der Matrizen $\boldsymbol{C}_z$, welche die Frequenzbedingungen Ω nach Abschnitt 5.4 definieren, erfolgt mithilfe des Computeralgebraprogramms Maple 2018.

$\boldsymbol{C}_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$	$\boldsymbol{C}_8 = \begin{bmatrix} 1 & 0 & -3 \end{bmatrix}$	$\boldsymbol{C}_{15} = \begin{bmatrix} 1 & -1 & -2 \end{bmatrix}$
$\boldsymbol{C}_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$	$\boldsymbol{C}_9 = \begin{bmatrix} 0 & 1 & -1 \end{bmatrix}$	$\boldsymbol{C}_{16} = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$
$\boldsymbol{C}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$	$\boldsymbol{C}_{10} = \begin{bmatrix} 1 & -1 & 2 \end{bmatrix}$	$\boldsymbol{C}_{17} = \begin{bmatrix} 1 & 1 & -2 \end{bmatrix}$
$\boldsymbol{C}_4 = \begin{bmatrix} 1 & 1 & 2 \end{bmatrix}$	$\boldsymbol{C}_{11} = \begin{bmatrix} 0 & 1 & -3 \end{bmatrix}$	$\boldsymbol{C}_{18} = \begin{bmatrix} 1 & 0 & 3 \end{bmatrix}$
$\boldsymbol{C}_5 = \begin{bmatrix} 0 & 1 & 3 \end{bmatrix}$	$\boldsymbol{C}_{12} = \begin{bmatrix} 1 & -1 & 0 \end{bmatrix}$	$\boldsymbol{C}_{19} = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$
$\boldsymbol{C}_6 = \begin{bmatrix} 1 & -\frac{1}{3} & 0 \end{bmatrix}$	$\boldsymbol{C}_{13} = \begin{bmatrix} 1 & \frac{1}{3} & 0 \end{bmatrix}$	$\boldsymbol{C}_{20} = \begin{bmatrix} 1 & -3 & 0 \end{bmatrix}$
$\boldsymbol{C}_7 = \begin{bmatrix} 1 & 3 & 0 \end{bmatrix}$	$\boldsymbol{C}_{14} = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$	$\boldsymbol{C}_{21} = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$

Tabelle 5.1: Matrizen $\boldsymbol{C}_z$ der Frequenzbedingungen Ω für das Simulationsbeispiel nach Abschnitt 5.7.

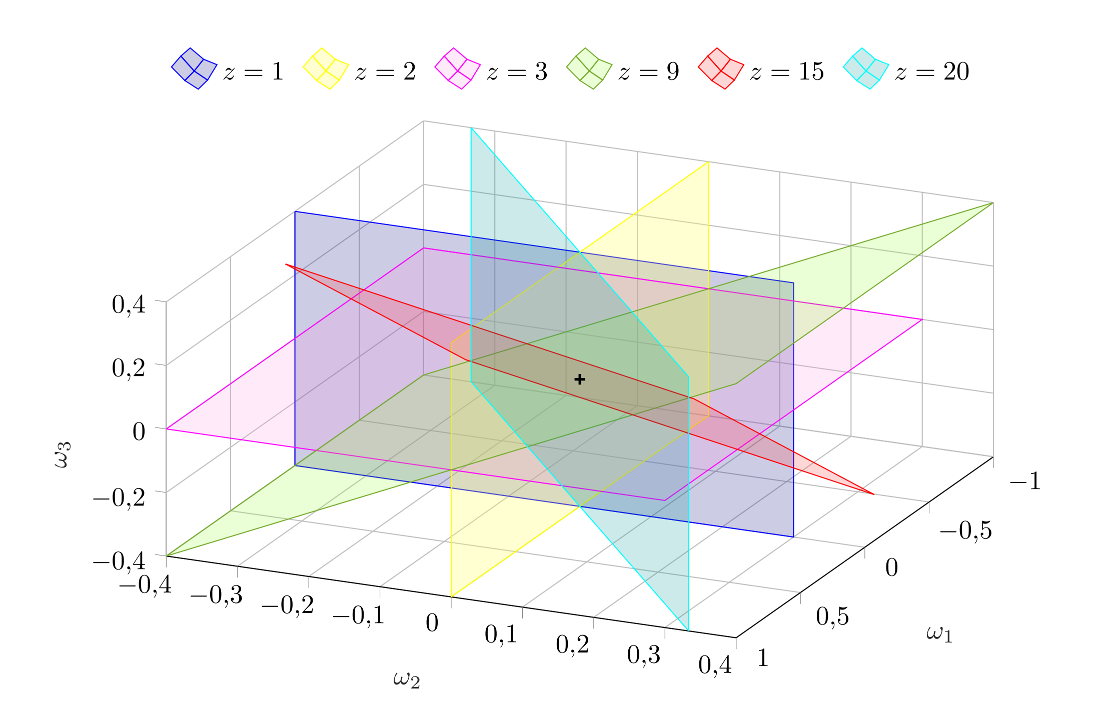


Abbildung 5.3: Grafische Veranschaulichung der Frequenzmenge Ω. Der Übersichtlichkeit halber wurden beispielhaft sechs der 21 Ebenen, welche durch $\boldsymbol{C}_z$ in Tabelle 5.1 definiert sind, und nach (5.89) verbotene Frequenzen beschreiben, skizziert. Der Koordinatenursprung ist durch das schwarze Kreuz gekennzeichnet.

5.7.2.2 Berechnung von $\boldsymbol{u}_{\mathrm{ex}}$ basierend auf $\boldsymbol{x}_{\mathrm{PE}}$

Um, wie in Abschnitt 5.6 beschrieben, mithilfe eines flachheitsbasierten Vorsteuerungsentwurfs das Anregungssignal $\boldsymbol{u}_{\mathrm{ex}}$ zu erhalten, wird zunächst das durch (5.85) gegebene Beispielsystem untersucht. Aufgrund von

$$\underbrace{\boldsymbol{g}_2(\boldsymbol{x})}_{=:\boldsymbol{g}_{\mathrm{II}}(\boldsymbol{x})} = \underbrace{\boldsymbol{g}_1(\boldsymbol{x})}_{=:\boldsymbol{g}_{\mathrm{I}}(\boldsymbol{x})} \underbrace{\frac{\sin\left(4x_1^2\right)+2}{\cos(2x_1)+2}}_{=:\gamma(\boldsymbol{x})} \tag{5.91}$$

(vgl. (5.68)) wird $\boldsymbol{g}_{\mathrm{I}}(\boldsymbol{x}) = \boldsymbol{g}_1(\boldsymbol{x})$ und $\boldsymbol{g}_{\mathrm{II}}(\boldsymbol{x}) = \boldsymbol{g}_2(\boldsymbol{x})$ gewählt und der zweite Eingang des Anregungssignals $\boldsymbol{u}_{\mathrm{ex}} = \begin{bmatrix} u_{\mathrm{ex},1} & u_{\mathrm{ex},2} \end{bmatrix}^{\mathsf{T}}$ wird zur Anregung nicht benötigt. Daher wird $u_{\mathrm{ex},2} = u_{\mathrm{ex,II}} = 0$ gesetzt und im Folgenden $u_{\mathrm{ex},1} = u_{\mathrm{ex,I}}$ betrachtet. Durch $h(\boldsymbol{x}) = x_1$ ist aufgrund von

$$\begin{aligned} L_{\boldsymbol{g}_{\mathrm{I}}} h(\boldsymbol{x}) &= 0 \\ L_{\boldsymbol{g}_{\mathrm{I}}} L_{\boldsymbol{f}_{\mathrm{g}}} h(\boldsymbol{x}) &= \cos(2x_1) + 2 =: D(\boldsymbol{x}) \end{aligned} \tag{5.92}$$

ein (fiktiver) Ausgang mit maximalem relativen Grad ($\delta = n = 2$) gegeben. Da außerdem $D(\boldsymbol{x}) \neq 0$ gilt, ist das betrachtete System flach, eine flachheitsbasierte Vorsteuerung existiert [Lév09, S. 143 f.], und aus dem Sollverlauf der Anregungstrajektorie (5.90) ergibt sich das Anregungssignal $u_{\mathrm{ex},1}$[113]. Die Gültigkeit von (5.79) in Annahme 5.6 kann, wie in Bemerkung 5.5 beschrieben, a priori bestätigt werden. Zudem gilt $\mathrm{Rang}(\boldsymbol{M}_i) = h_i$. Dies ist in Anhang C.3 ausgeführt. Somit resultiert eine Vorsteuerung der Form $u_{\mathrm{ex},1}\left(t, \hat{\boldsymbol{w}}_1^{[l]}, \hat{\boldsymbol{w}}_2^{[l]}\right)$, die eine exakte Systeminversion in Abhängigkeit der aktuellen Reglergewichte erlaubt.

5.7.3 Simulationsergebnisse

Neben den in Abschnitt 5.7.2 konstruierten Anregungssignalen $u_{\mathrm{ex},1,j}$, $j \in \mathbb{N}$, die aus unterschiedlichen Beispielsignalen $\bar{\boldsymbol{x}}_{\mathrm{PE},j}$ resultieren, soll der Einfluss weißen Rauschens als Anregungssignal untersucht werden. Dazu wird das Signal $u_{\mathrm{ex},1,\mathrm{w}}(t)$ verwendet, das in der Simulation durch eine Pseudozufallsfolge mit $\Delta t = 0{,}01\,\mathrm{s}$ und anschließendem Halteglied erzeugt wird, sodass eine mittelwertfreie weiße Gaußverteilung mit einer Standardabweichung von $2{,}8$ approximiert wird[114]. Im Folgenden wird die Konvergenz der Critic-Gewichte $\hat{\boldsymbol{w}}_i^{[l]}$, $i \in \mathcal{N}$, für diese unterschiedlichen Anregungssignale untersucht. Eine erste Untersuchung in Abschnitt 5.7.3.1 verwendet hierbei bezüglich der Maximalamplitude von $\boldsymbol{\sigma}_i$ normierte Lernraten. In einem zweiten Schritt werden in Abschnitt 5.7.3.2 Anregungssignale mit gleicher mittlerer Leistung bei Nutzung identischer Lernraten betrachtet.

[113] Die analytischen Berechnungen erfolgen mithilfe des Computeralgebraprogramms MAPLE 2018. Die anschließenden numerischen Simulationen erfolgen in MATLAB.

[114] Der Wert 2,8 der Standardabweichung ist so gewählt, dass die mittlere Signalleistung von $u_{\mathrm{ex},1,\mathrm{w}}(t)$ über der Simulationsdauer T_{sim} der mittleren Signalleistung von $u_{\mathrm{ex},1,1}(t)$ entspricht. Dies spielt im späteren Vergleich der Effizienz von Anregungssignalen derselben mittleren Signalleistung eine Rolle.

5.7.3.1 Normierte Lernraten

Um den Einfluss unterschiedlicher Signalamplituden von $\boldsymbol{\sigma}_i$ zu Vergleichbarkeitszwecken der verschiedenen Anregungssignale auszugleichen, werden in einer ersten Simulation normierte Lernraten

$$\eta_{\text{norm},i} = \frac{\eta_{\text{v}}}{\sigma_{i,\max}^2}, \quad \forall i \in \{1,2\}, \tag{5.93}$$

verwendet. Hierbei stellt $\eta_{\text{v}} > 0$ eine Konstante dar, die für alle Anregungssignale identisch gewählt ist, und $\sigma_{i,\max} \geq \|\boldsymbol{\sigma}_i\|_2$ wird simulativ für jedes der verwendeten Anregungssignale ermittelt. Mit der normierten Lernrate $\eta_{\text{norm},i}$ aus (5.93) folgt für die Abschätzung der Konvergenzrate

$$\rho_i = 1 - \frac{2T_i\eta_{\text{v}}}{(1+T_i\eta_{\text{v}})^2}\frac{\alpha_{\text{I},i}^2}{\sigma_{i,\max}^2}, \quad \forall i \in \{1,2\}, \tag{5.94}$$

(vgl. (5.22)). Da nach (5.19) größere Amplituden von $\boldsymbol{\sigma}_i$ und somit größere Werte von $\sigma_{i,\max}$ in gleichem Maße zu einer Erhöhung von $\alpha_{\text{I},i}$ führen, gleicht die Normierung der Lernrate unterschiedliche Signalamplituden von $\boldsymbol{\sigma}_i$ aus. Die Abschätzung der Konvergenzgeschwindigkeit ρ_i des Gradientenabstiegs nach (5.15) hängt (neben der für alle Anregungen identisch gewählten Größe η_{v}) somit nur noch von der Qualität der Erfüllung der PE-Bedingung ab, da ein höherer Grad $\alpha_{\text{I},i}$ der Erfüllung der PE-Bedingung (bei gleichbleibendem $\sigma_{i,\max}$ und gleicher Zeitkonstante T_i) zu schnellerer Konvergenz führt.

Im Folgenden werden die drei Beispielsignale

$$\begin{aligned}
\bar{\boldsymbol{x}}_{\text{PE},1} &= \frac{1}{4}\begin{bmatrix}\sin\left(1\,\text{s}^{-1}t\right)+\sin\left(2\,\text{s}^{-1}t\right)\\ \sin\left(3\,\text{s}^{-1}t\right)\end{bmatrix},\\
\bar{\boldsymbol{x}}_{\text{PE},2} &= \frac{1}{4}\begin{bmatrix}\sin\left(\frac{1}{2}\text{s}^{-1}t\right)+\sin\left(1\text{s}^{-1}t\right)\\ \sin\left(2\,\text{s}^{-1}t\right)\end{bmatrix},\\
\bar{\boldsymbol{x}}_{\text{PE},3} &= 2\bar{\boldsymbol{x}}_{\text{PE},2}
\end{aligned} \tag{5.95}$$

gewählt (vgl. (5.90)), wobei gemäß Tabelle 5.1 $\boldsymbol{\omega} \in \Omega$ gilt. Die zugehörigen Anregungssignale $u_{\text{ex},1,j}\left(t, \hat{\boldsymbol{w}}_1^{[l]}, \hat{\boldsymbol{w}}_2^{[l]}\right)$, $\forall j \in \{1,2,3\}$, werden damit, wie in Abschnitt 5.6 und Abschnitt 5.7.2.2 beschrieben, berechnet. Die initialen Critic-Gewichte werden zu

$$\hat{\boldsymbol{w}}_1^{[0]} = \begin{bmatrix}1{,}783\\-2{,}33\\2{,}215\end{bmatrix}, \qquad \hat{\boldsymbol{w}}_2^{[0]} = \begin{bmatrix}0{,}8916\\-1{,}165\\1{,}107\end{bmatrix} \tag{5.96}$$

gewählt. Die Regelgesetze $\hat{\boldsymbol{\mu}}_i^{[0]}(\boldsymbol{x})$, $\forall i \in \mathcal{N}$, werden aus den in (5.96) gegebenen Gewichten über den Zusammenhang (5.24) berechnet. Diese Initialisierung entspricht stabilisierenden initialen Regelgesetzen, die jedoch nicht der Nash-Lösung (vgl. (5.87)) entsprechen. Mit dem

exemplarisch gewählten Wert $\eta_\mathrm{v} = 10^4$ ergeben sich die in Tabelle 5.2 gezeigten normierten Lernraten $\eta_{\mathrm{norm},i}$, $\forall i \in \mathcal{N}$, nach (5.93). Der Policy-Evaluation-Schritt erfolgt mithilfe des durch (5.15) beschriebenen Gradientenabstiegs. Als Abbruchbedingung der Policy Evaluation wird im Folgenden

$$\left\| \hat{\boldsymbol{w}}_i^{[l]}(t-\tau) - \hat{\boldsymbol{w}}_i^{[l]}(t) \right\|_2 < 10^{-3}, \tag{5.97}$$

$\forall i \in \mathcal{N}$, mit dem Designparameter $\tau = 40\,\mathrm{s}$, verwendet[115]. Der Policy-Improvement-Schritt geschieht anschließend mithilfe des Critic-Gewichts $\hat{\boldsymbol{w}}_i^{[l+1]}$ des jeweiligen Spielers $i \in \mathcal{N}$ basierend auf (5.24).

Abbildung 5.4 zeigt den Verlauf der Critic-Gewichte $\hat{\boldsymbol{w}}_i^{[l]}$, $i \in \{1,2\}$, exemplarisch für die Verwendung der Anregungssignale $u_{\mathrm{ex},1,1}$ und $u_{\mathrm{ex},1,\mathrm{w}}$. Die resultierenden Fehlernormen $\|\boldsymbol{w}_i^* - \hat{\boldsymbol{w}}_i\|_2$, $i \in \{1,2\}$, für die Anregungssignale $u_{\mathrm{ex},1,j}$ mit $j \in \{1,2,3\}$ und $u_{\mathrm{ex},1,\mathrm{w}}$ sind in Abbildung 5.5 gezeigt. Für alle vier betrachteten Anregungssignale konvergieren die adaptiven Optimalregler gegen die Nash-Lösung. Die Konvergenzzeiten T_konv, ab denen $\|\tilde{\boldsymbol{w}}_i(t)\|_2 < 10^{-3}$, $\forall i \in \mathcal{N}$, gilt, sind Tabelle 5.2 zu entnehmen.

Das Eigenwertsignal $\lambda_{\min,2}(t)$ zur Überprüfung der PE-Eigenschaft nach Lemma 5.11, das für die vier verwendeten Anregungssignale resultiert, ist in Abbildung 5.6 gezeigt. Gemäß dem in Abschnitt 5.5 beschriebenen Verfahren können für den gewählten numerischen Schwellwert $\alpha_\mathrm{t} = 10^{-4}$ für alle vier Anregungssignale Zeitpunkte T_t gefunden werden, für welche die Ungleichung (5.55) mit $T = T_\mathrm{t}$ und $\alpha = \alpha_\mathrm{t}$ erfüllt ist. Diese Zeitkonstanten T_t sind in Tabelle 5.2 gegeben. Die Eigenwertsignale $\lambda_{\min,1}(t)$ in Abbildung 5.7 zeigen, dass Lemma 5.10 mit $\alpha = 10^{-4}$ und $T = T_\mathrm{t}$ nach Tabelle 5.2 erfüllt ist, und somit die Signale $\boldsymbol{\sigma}_i$, $\forall i \in \{1,2\}$, für die betrachteten Anregungssignale tatsächlich die PE-Bedingung erfüllen.

5.7.3.2 Gleiche mittlere Anregungssignalleistung

Während eine Normierung der Lernraten des Gradientenabstiegs nach (5.93) insbesondere Unterschiede der Amplitude von $\boldsymbol{\sigma}_i$ zu Vergleichszwecken ausgleicht, soll zudem ein weiteres

	$u_{\mathrm{ex},1,1}(t)$	$u_{\mathrm{ex},1,2}(t)$	$u_{\mathrm{ex},1,3}(t)$	$u_{\mathrm{ex},1,\mathrm{w}}(t)$
$\eta_{\mathrm{norm},i}$, $\forall i \in \{1,2\}$	36,50	80	5,99	41,49
T_konv	241 s	497 s	490 s	3311 s
T_t	1,14 s	3,61 s	1,54 s	11,56 s

Tabelle 5.2: Normierte Lernraten $\eta_{\mathrm{norm},i}$ nach (5.93), Konvergenzzeiten T_konv und Zeitpunkte T_t als Kandidaten für (5.55) mit $T = T_\mathrm{t}$ und $\alpha = \alpha_\mathrm{t} = 10^{-4}$ für die Anregungssignale $u_{\mathrm{ex},1,j}(t)$ ($\forall j \in \{1,2,3\}$) und $u_{\mathrm{ex},1,\mathrm{w}}(t)$.

[115] Grundsätzlich ist τ hierbei ein frei wählbarer Designparameter. Da nach Lemma 5.1 jedoch zunächst nur eine Verbesserung der geschätzten Critic-Gewichte $\hat{\boldsymbol{w}}_i^{[l+1]}$ für ein Intervall der Länge T_i gewährleistet wird, sollte τ nicht kleiner als T_i gewählt werden.

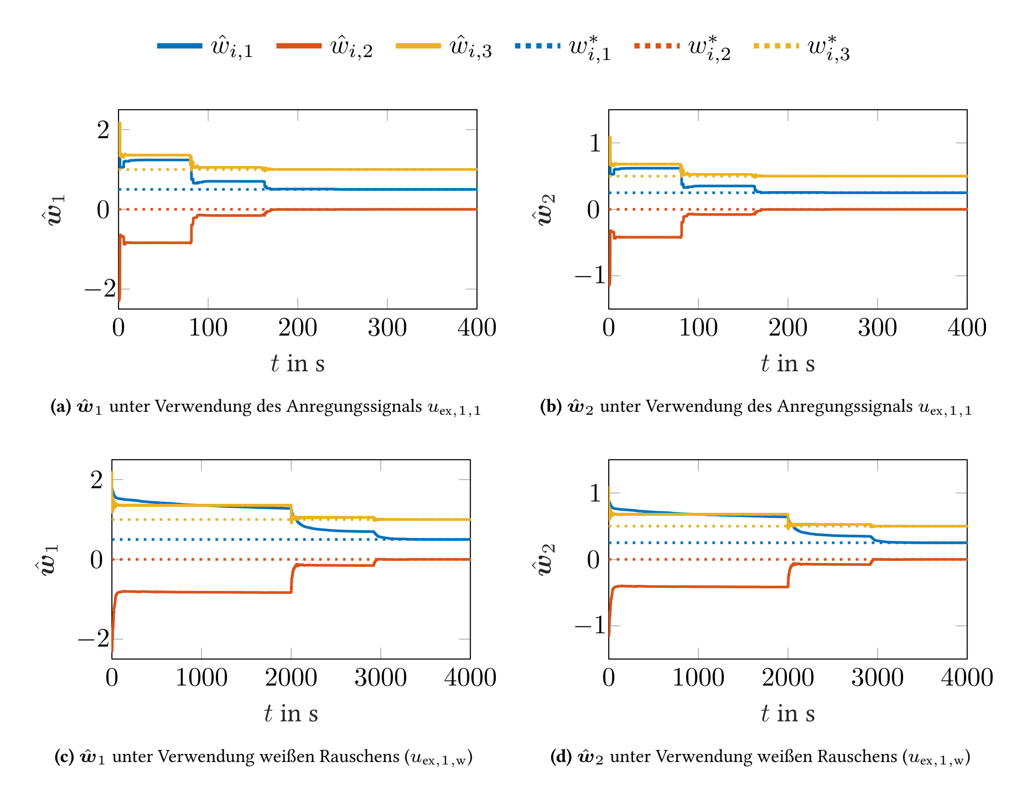


Abbildung 5.4: Verlauf der Critic-Gewichte $\hat{\boldsymbol{w}}_1$ und $\hat{\boldsymbol{w}}_2$ unter Verwendung des Anregungssignals $u_{\mathrm{ex},1,1}$ (oben) bzw. weißen Rauschens $u_{\mathrm{ex},1,\mathrm{w}}$ (unten) mit nach Tabelle 5.2 normierten Lernraten. Zu beachten ist insbesondere die um Faktor 10 unterschiedlich skalierte Zeitachse.

Kriterium zur Untersuchung der Effizienz der Anregungssignale betrachtet werden. Hierbei wird ein Anregungssignal als effizienter im Vergleich zu einem anderen Signal bezeichnet, wenn es bei identischer mittlerer Signalleistung

$$\frac{1}{T_{\mathrm{sim}}} \int_0^{T_{\mathrm{sim}}} \boldsymbol{u}_{\mathrm{ex}}^\mathsf{T}(t)\boldsymbol{u}_{\mathrm{ex}}(t)\ \mathrm{d}t \tag{5.98}$$

und gleicher Lernrate η_i zu schnellerer Konvergenz der Critic-Gewichte $\hat{\boldsymbol{w}}_i$ führt. Somit wird mit diesem Bewertungskriterium insbesondere bestraft, wenn ein Anregungssignal $\boldsymbol{u}_{\mathrm{ex}}(t)$ Signalanteile besitzt, die, beispielsweise durch eine mögliche Tiefpasscharakteristik des Systems, nicht zu einer Auslenkung des Systemzustands und somit zur Anregung von $\boldsymbol{\sigma}_i$ beitragen.

Daher sollen im Folgenden die Anregungssignale $u_{\mathrm{ex},1,1}$ und $u_{\mathrm{ex},1,\mathrm{w}}$, welche die gleiche mittlere Signalleistung nach (5.98) aufweisen und in Abbildung 5.8 gezeigt sind, bei identischer Lernrate $\eta_i = 1$, $i \in \mathcal{N}$, hinsichtlich der Konvergenz der Critic-Gewichte $\hat{\boldsymbol{w}}_i$ untersucht werden. Wie Abbildung 5.9 veranschaulicht, lenkt, bei gleicher mittlerer Signalleistung, das Anregungssignal $u_{\mathrm{ex},1,1}$ den Systemzustand $\boldsymbol{x}$ stärker aus als weißes Rauschen. Für diese

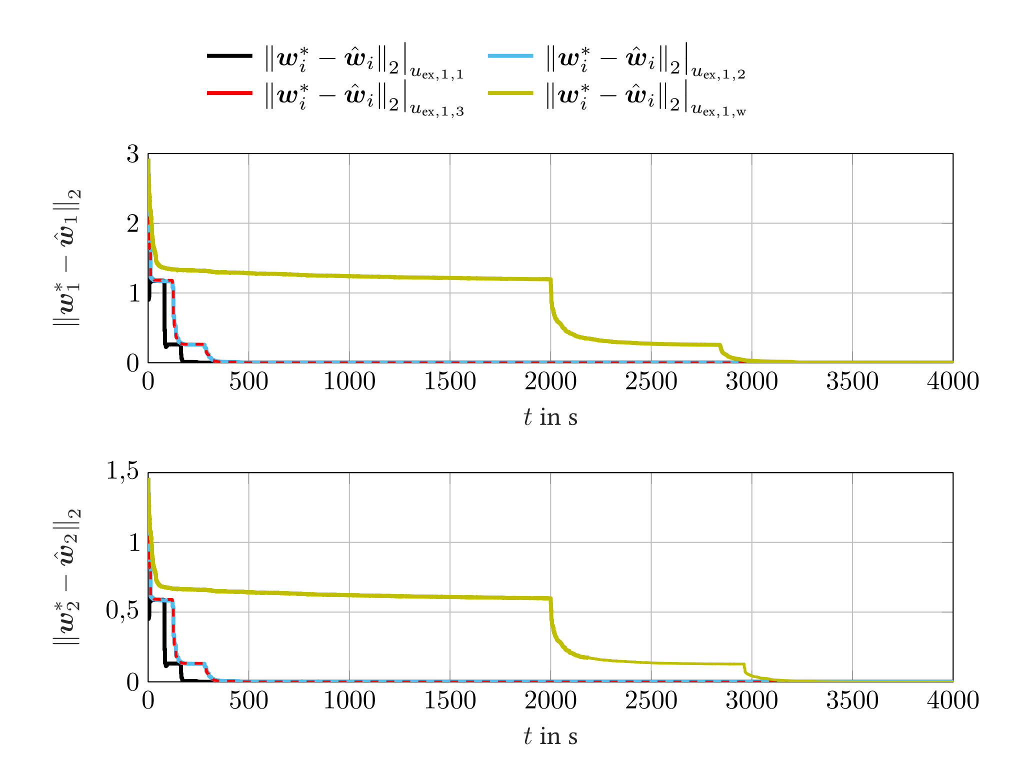


Abbildung 5.5: Norm des Fehlers der Critic-Gewichte $\|w_i^* - \hat{w}_i\|_2$, $i \in \{1, 2\}$, für die Anregungssignale $u_{ex,1,j}$, $j \in \{1, 2, 3\}$, (Sinussignale) und $u_{ex,1,w}$ (weißes Rauschen) mit nach Tabelle 5.2 normierten Lernraten.

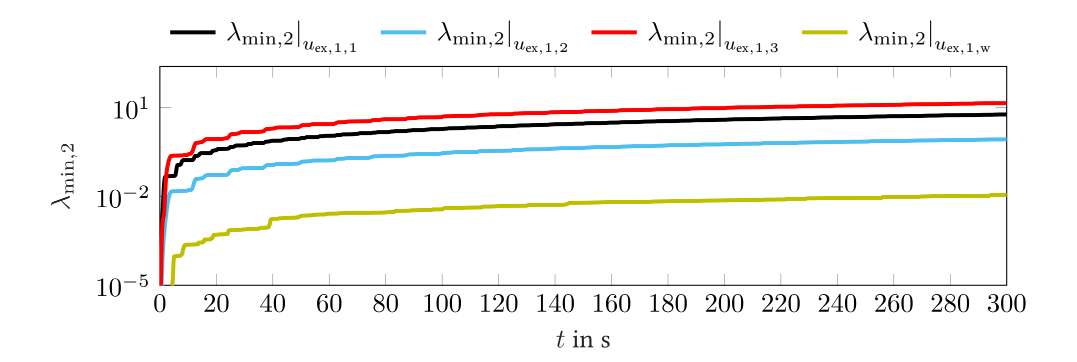


Abbildung 5.6: Eigenwertsignal $\lambda_{min,2}(t)$ zur Überprüfung der PE-Eigenschaft nach Lemma 5.11. Zu beachten ist die logarithmische Darstellung der Ordinate.

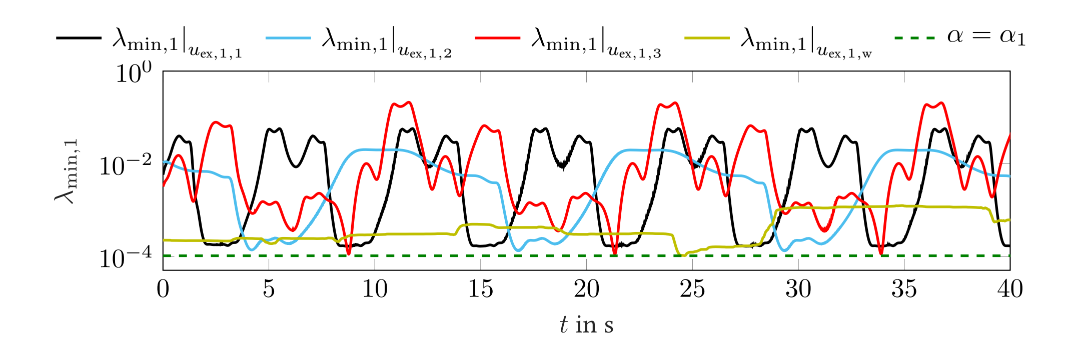


Abbildung 5.7: Eigenwertsignal $\lambda_{\min,1}(t)$ zur Überprüfung der PE-Eigenschaft nach Lemma 5.10. Der Übersichtlichkeit halber sind nur die ersten 30 s gezeigt. Die gestrichelte horizontale Linie entspricht dem gewählten Schwellwert $\alpha_\mathrm{t} = 10^{-4}$. Zu beachten ist die logarithmische Darstellung der Ordinate.

beiden Anregungssignale wird die Konvergenz eines einzelnen Policy-Evaluation-Schrittes im Folgenden betrachtet. Die initialen Critic-Gewichte $\hat{\boldsymbol{w}}_i^{[0]}$, $i \in \mathcal{N}$, und Regelgesetze $\hat{\boldsymbol{\mu}}_i^{[0]}(\boldsymbol{x})$ werden wie in Abschnitt 5.7.3.1 gewählt.

Der Verlauf der Critic-Gewichtsfehler $\left\|\boldsymbol{w}_i^{[1]} - \hat{\boldsymbol{w}}_i^{[1]}\right\|_2$, $i \in \mathcal{N}$, ist in Abbildung 5.10 gegeben. Bei der hier untersuchten Lernrate von $\eta_i = 1$ genügt die Simulationsdauer von 4000 s für die Anregung mit weißem Rauschen, d. h. unter Verwendung von $u_{\mathrm{ex},1,\mathrm{w}}$, nicht, um Konvergenz der Critic-Gewichte $\hat{\boldsymbol{w}}_i^{[l]}$ zu erreichen. So gilt in diesem Fall nach $t = 4000$ s

$$\left\|\boldsymbol{w}_1^{[1]} - \hat{\boldsymbol{w}}_1^{[1]}\right\|_2 = 0{,}31 \quad \text{und} \quad \left\|\boldsymbol{w}_2^{[1]} - \hat{\boldsymbol{w}}_2^{[1]}\right\|_2 = 0{,}16. \tag{5.99}$$

Unter Verwendung des Anregungssignals $u_{\mathrm{ex},1,1}$ folgt hingegen bereits nach $t = 338$ s, dass $\left\|\boldsymbol{w}_i^{[1]} - \hat{\boldsymbol{w}}_i^{[1]}\right\|_2 < 0{,}01$, $\forall i \in \{1, 2\}$, gilt.

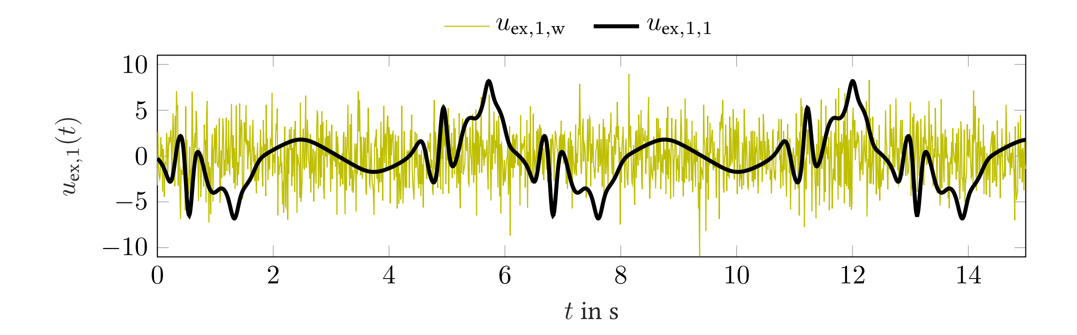


Abbildung 5.8: Anregungssignal $u_{\mathrm{ex},1}$ für $\bar{\boldsymbol{x}}_{\mathrm{PE},1}$ nach (5.95) (d. h. $u_{\mathrm{ex},1,1}$) und $u_{\mathrm{ex},1,\mathrm{w}}$ (d. h. weißes Rauschen) für die ersten 15 s der Simulation. Durch die Standardabweichung von 2,8 bei der Erzeugung von $u_{\mathrm{ex},1,\mathrm{w}}$ haben beide Anregungssignale die gleiche mittlere Signalleistung.

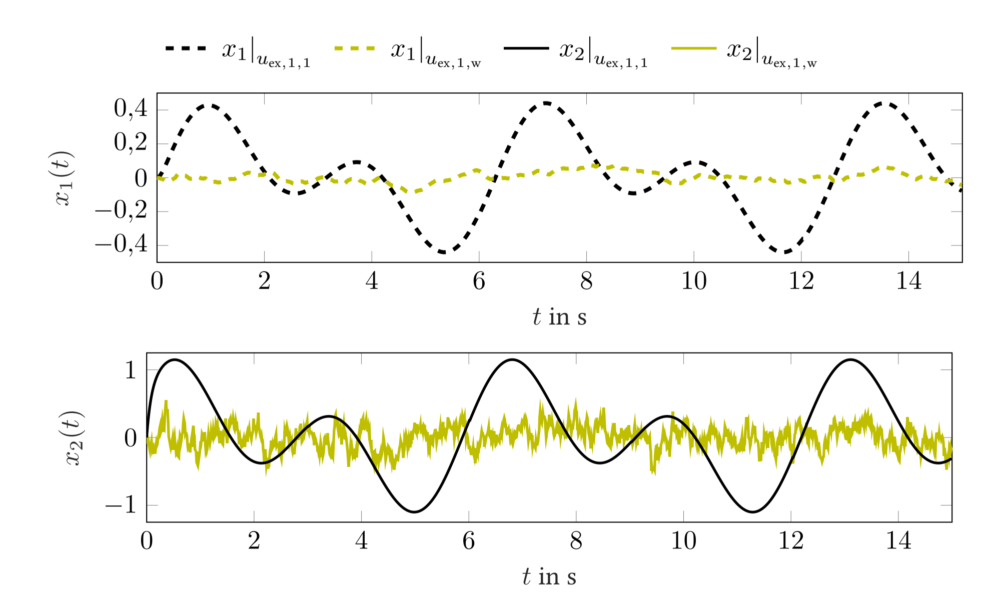


Abbildung 5.9: Verlauf der Zustandsgröße $\boldsymbol{x}(t)$ für die Anregungssignale $u_{\mathrm{ex},1,1}$ (Sinussignale) und $u_{\mathrm{ex},1,\mathrm{w}}$ (weißes Rauschen), welche die gleiche mittlere Anregungssignalleistung aufweisen.

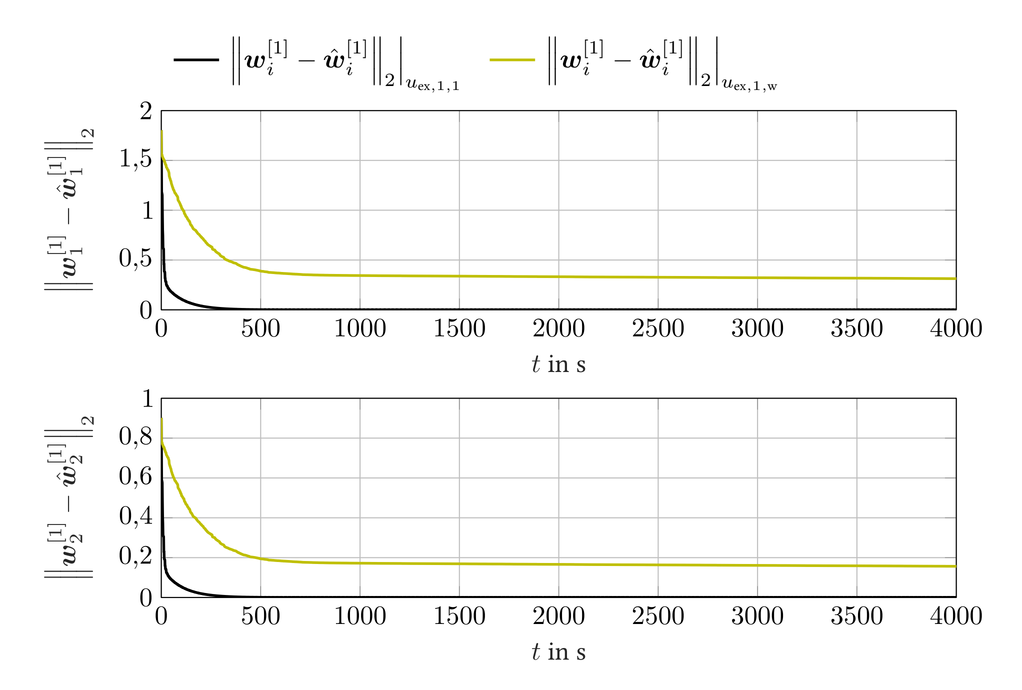


Abbildung 5.10: Verlauf der Critic-Gewichtsfehlernorm $\left\|\boldsymbol{w}_i^{[1]} - \hat{\boldsymbol{w}}_i^{[1]}\right\|_2$, $i \in \mathcal{N}$, unter Verwendung des Anregungssignals $u_{\mathrm{ex},1,1}$ bzw. weißen Rauschens $u_{\mathrm{ex},1,\mathrm{w}}$, welche die gleiche mittlere Anregungssignalleistung aufweisen, bei identischen Lernraten $\eta_i = 1$.

5.8 Diskussion

Die Erfüllung der PE-Eigenschaft spielt für die Konvergenz ADP-basierter Regelungsmethoden eine zentrale Rolle. Durch den nichtlinearen Zusammenhang zwischen dem Systemzustand $\boldsymbol{x}$ und den für den Adaptionsvorgang relevanten Signalen $\boldsymbol{\sigma}_i$ sind die in der Literatur vorhandenen theoretischen Aussagen zu linearen Transformationen von PE-Signalen nicht ausreichend, um Rückschlüsse darüber zu ziehen, für welche Zustandstrajektorien das Signal $\boldsymbol{\sigma}_i$ die PE-Eigenschaft erfüllt.

Die vorliegende Arbeit schließt diese Lücke, indem für polynomielle Basisfunktionen $\boldsymbol{\phi}(\boldsymbol{x})$ erstmalig ein allgemeingültiges hinreichendes Kriterium an den Systemzustand $\boldsymbol{x}$ präsentiert wird, sodass $\boldsymbol{\sigma}_i$ die PE-Bedingung erfüllt. Proposition 5.1 formuliert hierzu zunächst allgemeine Forderungen an eine Menge Ω an Frequenzbedingungen, um die Erfüllung der PE-Bedingung von $\boldsymbol{\sigma}_i$ zu gewährleisten. Durch Satz 5.2 ist schließlich eine konkrete Entwurfsmethode für eine solche Menge Ω gegeben. Damit gilt für jeden beliebigen Vektor $\boldsymbol{\omega} \in \Omega$, dass das Anregungstrajektoriensignal $\boldsymbol{x}_{\mathrm{PE}}(t)$ SR bezüglich $\dot{\boldsymbol{\phi}}(\cdot)$ und somit $\boldsymbol{\sigma} = \dot{\boldsymbol{\phi}}(\boldsymbol{x}_{\mathrm{PE}})$ PE ist. Da die genauen Amplituden der Sinus- und Kosinusterme in der Anregungstrajektorie $\boldsymbol{x}_{\mathrm{PE}}$ bei der Analyse nicht berücksichtigt wurden, sind die durch Ω beschriebenen Bedingungen (vgl. Proposition 5.1 und Satz 5.2) hinreichend. Somit können bei geeigneten Amplitudenverhältnissen gegebenenfalls $\boldsymbol{\omega} \notin \Omega$ existieren, für die $\boldsymbol{\sigma}_i$ dennoch die PE-Bedingung erfüllt. Für den praktischen Entwurf von Anregungssignalen ist dies jedoch unerheblich. Die Simulationsergebnisse bestätigen die Konvergenz einer gradientenbasierten Policy Evaluation, wenn $\boldsymbol{\sigma}_i \; \forall i \in \mathcal{N}$ die PE-Bedingung erfüllt. Dies gilt sowohl für eine Anregung mit weißem Rauschen als auch für Anregungssignale, die, wie in Abschnitt 5.6 beschrieben, auf Basis der theoretischen Erkenntnisse aus Satz 5.2 konstruiert sind. Für die praktische Überprüfung der Erfüllung der PE-Bedingung wurden in Abschnitt 5.5 zudem die Eigenwertsignale $\lambda_{\min,2}$ und $\lambda_{\min,1}$ präsentiert.

Die Untersuchungen des vorliegenden Kapitels beschränken sich aufgrund von Annahme 5.3 bislang auf polynomielle Basisfunktionen $\boldsymbol{\phi}(\boldsymbol{x})$. Für den Fall, dass diese Annahme verletzt ist, könnte eine Taylor-Approximation von $\boldsymbol{\phi}(\boldsymbol{x})$ zukünftig weitere Analysen ermöglichen. Bei hinreichend kleinem Restterm dieser Approximation könnten dann, ähnlich wie in Lemma 5.12, Rückschlüsse über die Erfüllung der PE-Bedingung gezogen werden. Auch sind die theoretischen Analysen bislang auf den Fall der zeitkontinuierlichen PE-Bedingung (2.27) begrenzt. Sie könnten jedoch als Ausgangspunkt für die Untersuchung der zeitdiskreten PE-Bedingung (2.29) dienen.

Aus den theoretischen Analysen dieses Kapitels lassen sich schließlich wesentliche Erkenntnisse schlussfolgern, die für den Entwurf von Anregungssignalen für ADP-basierte Regler hilfreich sind. So kann bereits eine geringe Anzahl an Frequenzen $\boldsymbol{\omega} \in \Omega$ im Systemzustand $\boldsymbol{x}$ ausreichend sein, damit $\boldsymbol{\sigma}_i$ PE ist. Hierbei lässt Satz 5.2 Freiheiten bei der Wahl der Frequenzen und Amplituden in $\boldsymbol{x}_{\mathrm{PE}}(t)$. Anstatt, wie bei der Anregung mit weißem Rauschen, alle Frequenzen gleich stark im Anregungssignal zu verwenden, können diese Freiheitsgrade dazu dienen,

die Anregungssignalleistung effizient zu nutzen. Auch könnten die Freiheitsgrade gegebenenfalls für eine zukünftige Optimierung der Konvergenzeigenschaften verwendet werden. Liegt darüber hinaus sogar (ggf. teilweise) Vorwissen über das Übertragungsverhalten des betrachteten Systems vor (beispielsweise über dessen Tiefpass- oder Bandpasscharakteristik), so können die Anregungsfrequenzen entsprechend gewählt werden.

Zusammenfassend verdeutlicht dieses Kapitel, dass die präsentierten Anregungssignale, die ihre Signalleistung auf niedrigere Bereiche des Frequenzspektrums konzentrieren, deutlich effizienter zur Anregung beitragen können, als dies bei der Verwendung von weißem Rauschen der Fall ist. Dies zeigt sich selbst bei der Nutzung normierter Lernraten durch eine wesentlich schnellere Konvergenz der Critic-Gewichte und somit des Policy-Iteration-Algorithmus ($85{,}0\,\%$–$92{,}7\,\%$ Verbesserung, vgl. Tabelle 5.2 und Abbildung 5.5). Bei gleicher mittlerer Anregungssignalleistung nach (5.98) zeigt sich bei identischen Lernraten $\eta_i = 1$ im betrachteten Simulationsbeispiel die Überlegenheit des analytisch berechneten Anregungssignals im Vergleich zu weißem Rauschen sogar noch deutlicher (vgl. Abbildung 5.10). Somit liefert dieses Kapitel Antworten auf die in Abschnitt 2.4.2 formulierte Forschungsfrage 2 hinsichtlich der Bedingungen an Systemzustände, um die PE-Eigenschaft zu erfüllen und Konvergenz der betrachteten ADP-Methode zu gewährleisten.

6 Reale Anwendung ADP-basierter Solltrajektorienfolgeregler

Nach den theoretischen Beiträgen zu zeitdiskreten (Kapitel 3) und zeitkontinuierlichen (Kapitel 4) ADP-kompatiblen Solltrajektoriendarstellungen und Konvergenzbedingungen (Kapitel 5) werden in diesem Kapitel zwei reale Anwendungsbeispiele ADP-basierter Solltrajektorienfolgeregler betrachtet. Da bei diesen realen Systemen Messungen der Ausgangs- bzw. Zustandsgrößen zu diskreten Zeitpunkten stattfinden, wird eine zeitdiskrete Darstellung verwendet. Unter den in Kapitel 3 präsentierten zeitdiskreten Methoden erscheint insbesondere die in Abschnitt 3.2 vorgestellte ADP-kompatible parametrierte Referenztrajektoriendarstellung vielversprechend, da durch die parametrische lokale Approximation des Solltrajektorienverlaufs eine kompakte Darstellung vorliegt (vgl. auch Abschnitt 3.4).

In Abschnitt 6.1 wird ein modellfreier Actor-Critic-Ansatz angewandt, um eine Längsregelung eines realen Fahrzeugs zu realisieren, wobei online, d. h. während der Fahrt, die Reglergewichte adaptiert werden. Motiviert ist dieses Anwendungsbeispiel durch eine wachsende Modellvielfalt und den Wunsch, ausgehend von einer initialen Reglerparametrierung, die beispielsweise aus einem vorhandenen modellbasierten Reglerentwurf resultiert, eine automatisierte Feinabstimmung vornehmen zu können. Durch die Verwendung der parametrischen Darstellung des Sollgeschwindigkeitsverlaufs wird dabei zusätzlich zu einem Ausgangsrückführungsterm ein Vorsteuerterm gelernt. Abschnitt 6.2 betrachtet schließlich ein reales Ball-auf-Platte-System. Für dieses beliebte regelungstechnische Benchmarksystem wird ebenfalls ein modellfreier Trajektorienfolgeregler trainiert und mit einem modellbasierten Ansatz verglichen. Betrachtet werden dabei modellfreie ADP-Solltrajektorienfolgeregler, die den Sollpositionsverlauf einerseits durch ein Polynom zweiten Grades und andererseits durch eine stationäre Sollzustandsvorgabe beschreiben, sowie ein modellbasierter Vergleichsregler. Hierbei offenbaren sich die Vorteile des präsentierten Ansatzes, der ohne aufwendige Modellbildung zu den geringsten Kosten im Sinne des zugrunde liegenden Gütefunktionals führt.

6.1 Modellfreie, adaptive Längsregelung eines realen Fahrzeugs

In diesem Abschnitt wird die in Kapitel 3 vorgestellte ADP-kompatible parametrierte Referenztrajektoriendarstellung verwendet, um einen Geschwindigkeitsregler, der einem vorgegebenen

Geschwindigkeitsprofil folgen soll, mithilfe einer Online-ADP-Methode in einem realen Fahrzeug zu trainieren[116]. Das verwendete Versuchsfahrzeug ist in Abbildung 6.1 zu sehen.

Bei der ADP-basierten Geschwindigkeitsregelung eines realen Fahrzeugs ergeben sich zwei wesentliche Herausforderungen. Erstens muss der Sollgeschwindigkeitsverlauf durch eine ADP-kompatible Solltrajektoriendarstellung repräsentiert und in den ADP-Formalismus integriert werden (vgl. Abschnitt 2.2 und Kapitel 3). Zweitens basiert die Längsdynamik des Versuchsfahrzeugs auf einer Aktuatorik (Antriebsstrang und Bremssystem) mit vergleichsweise langsamen dynamischen Vorgängen und teilweise nicht gemessenen internen Zuständen (vgl. [PRH19]). Anstelle des Systemzustands liegt nur eine Ausgangsgröße in Form der gemessenen Fahrzeuggeschwindigkeit vor. Würden die nicht messbaren Zustände der Aktuatordynamik jedoch schlicht ignoriert, wäre damit die Markov-Annahme verletzt. Die in der Literatur vorhandenen ADP- oder RL-basierten Geschwindigkeitsregler, die zudem meist auf Simulationen basieren oder nicht online lernen (vgl. Abschnitt 2.2.4), ignorieren dieses Problem bislang. Beispielhaft seien [BK18], [NCH08], [DCd11], [PT12], [WXL+14] und [WZLD15] zu nennen. Im Folgenden wird hingegen, wie in den Arbeiten von Puccetti et al. [PRH19], [PKRH20] vorgestellt und in Anhang D.1 motiviert, ein FIR-Filter[117] verwendet, um den Lernvorgang durch einen geschätzten Hilfszustand zu stützen.

Die zentralen Ideen sind somit die Verwendung der in Abschnitt 3.2 vorgestellten zeitdiskreten ADP-kompatiblen parametrierten Referenztrajektorie sowie eines FIR-Filters zur Rekonstruktion fehlender Zustandsinformationen. Ersteres ermöglicht, nicht nur die aktuelle Wunschgeschwindigkeit, sondern eine lokale Approximation des aktuellen Wunschgeschwindigkeitspro-



Abbildung 6.1: Versuchsfahrzeug (*BMW 740Li*), das für die modellfreie, adaptive Längsregelung mittels eines Online-ADP-Ansatzes verwendet wurde. Bildquelle: *BMW Press Club* (Zugriff am 19.10.2021).

[116] Ergebnisse dieses Abschnitts wurden im Rahmen zweier Konferenzbeiträge veröffentlicht [KPRH20], [PKRH20].

[117] (engl.): *finite impulse response filter*. Filter mit endlicher Impulsantwort.

fils explizit zu berücksichtigen. Die Off-Policy-Charakteristik (vgl. Abschnitt 2.1.4.4) der verwendeten Actor-Critic-Methode erlaubt zudem einerseits, das Wunschgeschwindigkeitsprofil während des Trainingsvorgangs künstlich durch zusätzliches Anregungsrauschen des Sollverlaufs zu überlagern, und andererseits die Verwendung von Experience Replay [MKS+13], [MKS+15].

Nachfolgend wird zunächst die zugrunde liegende Problemstellung formuliert, bevor der verwendete modellfreie ADP-Solltrajektorienfolgeregelungsalgorithmus mit Zustandsrekonstruktion vorgestellt wird. Anschließend werden Trainingsergebnisse im Realfahrzeug sowie Validierungsfahrten präsentiert. Eine abschließende Diskussion rundet den Abschnitt ab.

6.1.1 Problemstellung

Ziel ist ein datenbasierter, selbstlernender ADP-Algorithmus, der in einem Realfahrzeug umgesetzt wird und online einen Geschwindigkeitsregler adaptiert, der einer ADP-kompatiblen Approximation gewünschter Sollgeschwindigkeitsverläufe optimal im Sinne eines gegebenen Gütefunktionals folgt.

Der ADP-Regler verfügt dabei jedoch über kein Modell der Longitudinaldynamik, die Antriebsstrang, Bremssystem und eine unterlagerte Regelung beinhaltet. Wie in Abbildung 6.2 dargestellt, beschreibt diese Longitudinaldynamik den Zusammenhang zwischen der Eingangsgröße u_k, die der angeforderten Beschleunigung im Zeitschritt k entspricht, und der tatsächlichen Beschleunigung a_k, die jedoch nicht gemessen wird[118]. Stattdessen steht dem ADP-Regler lediglich eine Messung der Geschwindigkeit y_k zur Verfügung.

Zum Zeitschritt k ist der Sollgeschwindigkeitsverlauf auf einem Vorausschauhorizont der Länge n_h durch $y_{\mathrm{r,soll},k}, y_{\mathrm{r,soll},k+1}, \dots, y_{\mathrm{r,soll},k+n_\mathrm{h}-1}$ beschrieben. Aus diesem zunächst beliebigen Geschwindigkeitsprofil wird in jedem Zeitschritt k eine nach Definition 3.1 ADP-kompatible lokale Approximation

$$y_\mathrm{r}\left(\boldsymbol{z}_k^{(\kappa)}\right) := y_\mathrm{r}(\boldsymbol{z}_k, \kappa) = \boldsymbol{z}_k^\intercal \boldsymbol{\rho}(\kappa) \tag{6.1}$$

(vgl. (3.15)) des Solltrajektorienverlaufs berechnet. Dabei gewichtet $\boldsymbol{z}_k$ die gegebenen Basisfunktionen $\boldsymbol{\rho}(\kappa)$ und κ bezeichnet einen Zeitindex. Gesucht ist ein Regelgesetz $\mu\left(y_\kappa, \boldsymbol{z}_k^{(\kappa)}\right)$, das die Value Function

$$\begin{aligned} V^\mu\left(y_k, \boldsymbol{z}_k\right) &= \sum_{\kappa=0}^{\infty} \gamma^\kappa r\left(y_{k+\kappa}, y_\mathrm{r}\left(\boldsymbol{z}_k^{(\kappa)}\right), \mu\left(y_{k+\kappa}, \boldsymbol{z}_k^{(\kappa)}\right)\right) \\ &= \sum_{\kappa=0}^{\infty} \gamma^\kappa \left(Q_\mathrm{y}\left(y_{k+\kappa} - y_\mathrm{r}\left(\boldsymbol{z}_k^{(\kappa)}\right)\right)^2 + R\left(\mu\left(y_{k+\kappa}, \boldsymbol{z}_k^{(\kappa)}\right)\right)^2\right) \end{aligned} \tag{6.2}$$

[118] Abhängig vom Entwurfsziel des Reglers und anderen Faktoren, wie beispielsweise äußeren Störeinflüssen, kann es sinnvoll sein, dass die Stellgröße u_k von der numerischen Differenziation des Sollgeschwindigkeitsprofils abweicht. Das Ziel des ADP-Reglers ist daher, die optimale Stellgröße u_k bezüglich eines vorgegebenen Gütefunktionals zu erlernen.

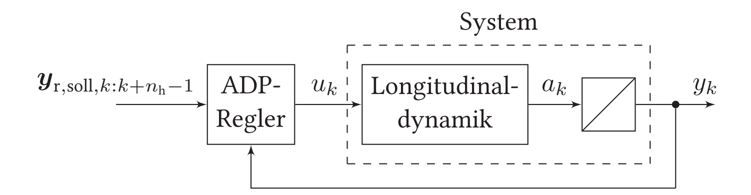


Abbildung 6.2: Struktur der gegebenen Problemstellung zur Längsregelung eines realen Fahrzeugs. Basierend auf der Ausgangsgröße y_k, die der gemessenen Geschwindigkeit des Fahrzeugs entspricht, sowie dem Sollgeschwindigkeitsprofil $\boldsymbol{y}_{\mathrm{r,soll},k:k+n_{\mathrm{h}}-1} := \begin{bmatrix} y_{\mathrm{r,soll},k} & y_{\mathrm{r,soll},k+1} & \cdots & y_{\mathrm{r,soll},k+n_{\mathrm{h}}-1} \end{bmatrix}$ und einem internen Kostensignal $r(\cdot)$ soll der Regler adaptiert werden.

minimiert (vgl. (3.37)). Hierbei wird mit $Q_{\mathrm{y}} > 0$ und $R > 0$ sowohl die Abweichung der Geschwindigkeit $y_{k+\kappa}$ im Zeitschritt $k+\kappa$ von der approximierten Sollgeschwindigkeit $y_{\mathrm{r}}\left(\boldsymbol{z}_k^{(\kappa)}\right)$ als auch der Stellaufwand quadratisch bestraft. Eine im Sinne der ADP-kompatiblen Solltrajektoriendarstellung resultierende Verschiebung des Parametervektors $\boldsymbol{z}_k$ um κ Zeitschritte ist dabei durch

$$\boldsymbol{z}_k^{(\kappa)} = \boldsymbol{z}_k \boldsymbol{D}(\kappa) \tag{6.3}$$

(vgl. Definition 3.2) gegeben. $\boldsymbol{D}(\kappa)$ stellt eine Verschiebungsmatrix dar, die so gewählt wird, dass

$$y_{\mathrm{r}}\left(\boldsymbol{z}_k^{(\kappa+j)}\right) = y_{\mathrm{r}}\left(\boldsymbol{z}_k^{(\kappa)}, j\right) = y_{\mathrm{r}}(\boldsymbol{z}_k, \kappa+j), \quad \forall \kappa, j \in \mathbb{N}_{\geq 0}, \tag{6.4}$$

gilt.

Bemerkung 6.1
Die ADP-kompatible Approximation des Sollgeschwindigkeitsverlaufs in (6.1) *durch den endlichdimensionalen Vektor $\boldsymbol{z}_k$ hat zur Folge, dass die Value Function V^{μ} in* (6.2) *trotz des unendlichen Optimierungshorizonts durch einen funktionalen Zusammenhang der Form $V^{\mu}(y_k, \boldsymbol{z}_k)$ beschrieben werden kann (vgl. Proposition 3.1).*

6.1.2 Modellfreier ADP-Solltrajektorienfolgeregler mit Zustandsrekonstruktion

In diesem Abschnitt wird der verwendete modellfreie ADP-Solltrajektorienfolgeregler mit Zustandsrekonstruktion und Vorsteuerterm präsentiert. Grundsätzlich finden hierbei zwei wesentliche Modifikationen statt. Erstens wird in Abschnitt 6.1.2.1 eine erweiterte Q-Function definiert. Neben der gemessenen Geschwindigkeit y_k hängt diese Q-Function vom Vektor $\boldsymbol{z}_k$ ab, der im Zeitschritt k den approximierten Sollgeschwindigkeitsverlauf beschreibt (vgl. (6.1)). Zweitens werden dieser Q-Function zusätzlich vergangene Stellgrößen zur Verfügung gestellt, um, wie in Abschnitt 6.1.2.2 beschrieben, einen Hilfszustand $\bar{x}_k$ zu schätzen, der den

Lernprozess stützt[119]. Die Schätzung dieses Hilfszustands erfolgt mithilfe eines FIR-Filters. Schließlich wird in Abschnitt 6.1.2.3 eine Übersicht über den verwendeten Algorithmus des ADP-basierten Geschwindigkeitsreglers präsentiert.

6.1.2.1 Q-Function und Solltrajektorienapproximation

Die zur Value Function V^{μ} nach (6.2) gehörende Q-Function ergibt sich zu[120]

$$Q^{\mu}\left(y_k, \boldsymbol{z}_k, u_k\right) = r\Big(y_k, y_{\mathrm{r}}(\boldsymbol{z}_k), u_k\Big) + \gamma Q^{\mu}\left(y_{k+1}, \boldsymbol{z}_k^{(1)}, \mu\Big(y_{k+1}, \boldsymbol{z}_k^{(1)}\Big)\right). \tag{6.5}$$

Der zu trainierende Geschwindigkeitsregler basiert im Folgenden auf einer Approximation erster Ordnung (Polynom vom Grad $d = 1$) des Sollgeschwindigkeitsprofils. Somit gilt

$$\boldsymbol{\rho}(\kappa) = \begin{bmatrix} \kappa\Delta t \\ 1 \end{bmatrix}, \quad \boldsymbol{D}(\kappa) = \begin{bmatrix} 1 & \kappa\Delta t \\ 0 & 1 \end{bmatrix}, \tag{6.6}$$

wobei Δt die Abtastzeit des Systems beschreibt[121]. Zu Vergleichszwecken[122] wird ein zweiter Regler betrachtet, der in jedem Zeitschritt k den Sollgeschwindigkeitsverlauf durch eine konstante Approximation (Polynom vom Grad $d = 0$) beschreibt, d. h.

$$\boldsymbol{\rho}(\kappa) = 1, \quad \boldsymbol{D}(\kappa) = 1. \tag{6.7}$$

Analog zu (3.72) mit $\beta = 1$ wird der Referenztrajektorienparameter $\boldsymbol{z}_k$ durch

$$\boldsymbol{z}_k^{\mathsf{T}} = \begin{bmatrix} z_{1,k} & \dots & z_{d+1,k} \end{bmatrix} = \begin{bmatrix} y_{\mathrm{r,soll},k} & y_{\mathrm{r,soll},k+1} & \dots & y_{\mathrm{r,soll},k+n_{\mathrm{h}}-1} \end{bmatrix} \boldsymbol{P}_{\mathrm{LS}} \tag{6.8}$$

berechnet. Hierbei kann die durch

$$\boldsymbol{P}_{\mathrm{LS}} \coloneqq \begin{bmatrix} \boldsymbol{\rho}^{\mathsf{T}}(0) \\ \boldsymbol{\rho}^{\mathsf{T}}(1) \\ \vdots \\ \boldsymbol{\rho}^{\mathsf{T}}(n_{\mathrm{h}}-1) \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{\rho}^{\mathsf{T}}(0) \\ \boldsymbol{\rho}^{\mathsf{T}}(1) \\ \vdots \\ \boldsymbol{\rho}^{\mathsf{T}}(n_{\mathrm{h}}-1) \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \boldsymbol{\rho}^{\mathsf{T}}(0) \\ \boldsymbol{\rho}^{\mathsf{T}}(1) \\ \vdots \\ \boldsymbol{\rho}^{\mathsf{T}}(n_{\mathrm{h}}-1) \end{bmatrix} \right)^{-1} \tag{6.9}$$

---

[119] Dieser Hilfszustand muss dabei nicht notwendigerweise physikalisch interpretierbar sein. Neben der nicht gemessenen Beschleunigung a_k können weitere interne Größen der Longitudinaldynamik Einfluss auf $\bar{x}_k$ haben.

[120] Grundsätzlich kann sich der Parameter $\boldsymbol{z}_k$, der den Solltrajektorienverlauf beschreibt, beliebig mit der Zeit k ändern. Daher würde die Verwendung von $\boldsymbol{z}_k$ und $\boldsymbol{z}_{k+1}$ in (6.5) die Markov-Annahme verletzen. Aufgrund der Verschiebungsmatrix $\boldsymbol{D}(\kappa)$ in (6.3), welche die durch $\boldsymbol{z}_k$ gegebene Approximation des Solltrajektorienverlaufs um einen Zeitschritt propagiert, ist die Markov-Annahme für den Zustandsübergang von $\begin{bmatrix} y_k & \boldsymbol{z}_k^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$ nach $\begin{bmatrix} y_{k+1} & \boldsymbol{z}_k^{(1)\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$ gewährleistet. Deshalb wird in (6.5) $\boldsymbol{z}_k^{(1)}$ verwendet (vgl. auch Bemerkung 3.2).

[121] Die verwendete Approximation erster Ordnung hat sich für dieses Anwendungsbeispiel als weniger anfällig gegenüber Oszillationen erwiesen und der zugehörige ADP-Regler ist zudem im Vergleich zu Approximatoren höherer Ordnung einfacher zu trainieren – insbesondere im Hinblick auf die zusätzlich benötigte Zustandsrekonstruktion durch ein FIR-Filter.

[122] Ziel der vorliegenden Arbeit ist das Aufzeigen der grundsätzlichen Anwendbarkeit der vorgestellten Methode. Für eine ausführliche Diskussion des hier betrachteten Anwendungsbeispiels sei auf die Arbeiten von Puccetti et al. [PRH19], [PKRH20], [KPRH20] verwiesen.

definierte Projektionsmatrix aufgrund der Unabhängigkeit von $y_{\mathrm{r,soll}}$ einmalig im Voraus berechnet und gespeichert werden, um Rechenzeit während des Onlinetrainings und Onlinebetriebs einzusparen. Der Rechenaufwand für die Approximation des Sollverlaufs nach (3.72) reduziert sich somit in jedem Zeitschritt k zu einer einzelnen Matrixmultiplikation.

Bemerkung 6.2

Ein Regler, der auf einer konstanten Sollgeschwindigkeitsapproximation ($d = 0$) basiert, entspricht dem in [PKRH20] präsentierten Ansatz zur ADP-basierten Geschwindigkeitsregelung ohne Vorsteuerterm. Die Verwendung einer linearen Sollgeschwindigkeitsapproximation ($d = 1$) hat hingegen zur Folge, dass dem Regler zusätzlich zur aktuellen Sollgeschwindigkeit die aktuelle Sollbeschleunigung, d. h. die erste Ableitung des Sollgeschwindigkeitsprofils im Zeitschritt k, zur Verfügung gestellt wird. Die Projektion in (6.8) *übernimmt dabei unter anderem die Aufgabe einer numerischen Differenziation*[123]*.*

6.1.2.2 Netzwerkarchitektur der Q-Function-Approximation

Dem ADP-Regler werden, neben y_k und $\boldsymbol{z}_k$, vergangene Stellgrößen übergeben, aus denen fehlende Zustandsinformation rekonstruiert werden soll. Diese Idee folgt damit dem Vorgehen von Puccetti et al. [PKRH20], [PRH19]. Konkret wird eine Approximation

$$\bar{x}_k \coloneqq \boldsymbol{w}_{\mathrm{FIR}}^{\mathsf{T}} \boldsymbol{u}_{k-h_{\mathrm{FIR}}:k-1} \tag{6.10}$$

aus h_{FIR} vergangenen Stellgrößen

$$\boldsymbol{u}_{k-h_{\mathrm{FIR}}:k-1} \coloneqq \begin{bmatrix} u_{k-h_{\mathrm{FIR}}} & u_{k-h_{\mathrm{FIR}}+1} & \cdots & u_{k-1} \end{bmatrix}^{\mathsf{T}} \tag{6.11}$$

und den zu schätzenden Gewichten $\boldsymbol{w}_{\mathrm{FIR}} \in \mathbb{R}^{h_{\mathrm{FIR}}}$ verwendet. Dies entspricht einem FIR-Filter. Eine geeignete Länge h_{FIR} dieses Filters ergibt sich grundsätzlich anhand der Einschwingzeit der Impulsantwort der Aktuatordynamik (vgl. [PRH19]). Basierend auf Erfahrungswerten hat sich die Rekonstruktion eines einzelnen Zustands, wie in (6.10) gezeigt, für das hier betrachtete Beispiel der Longitudinaldynamik bewährt [PRH19, Abschnitt VI].

Die verwendete Netzwerkarchitektur, die diesen Rekonstruktionsmechanismus und den durch $\boldsymbol{z}_k$ approximierten Sollgeschwindigkeitsverlauf integriert, ist in Abbildung 6.3 gegeben. Die Eingangsgrößen sind durch die Geschwindigkeit y_k, den Referenzparametervektor $\boldsymbol{z}_k$, vergangene Stellgrößen $\boldsymbol{u}_{k-h_{\mathrm{FIR}}:k-1}$ sowie die aktuelle Stellgröße u_k gegeben. Die Schätzung $Q_{\boldsymbol{w}}^{\mu_{\boldsymbol{\theta}}}(\tilde{\boldsymbol{x}}_k, u_k)$ der Q-Function $Q^{\mu}(\tilde{\boldsymbol{x}}_k, u_k)$ bildet die Ausgangsgröße. Die Wahl der quadratischen Schicht (‚qf') wird in der Arbeit von Puccetti et al. [PRH19] motiviert und folgt aufgrund näherungsweise linearer Dynamik des Systems sowie der quadratischen Form des Gütefunktionals nach (6.2). Im nächsten Abschnitt folgt die Vorstellung des verwendeten Reglers sowie dessen Trainingsprozedur.

[123] Beispielsweise resultiert für den Fall $d = 1$ und $n_{\mathrm{h}} = 2$ aus (6.9) gerade $\boldsymbol{P}_{\mathrm{LS}} = \begin{bmatrix} -\Delta t & 1 \\ \Delta t & 0 \end{bmatrix}$ und somit aus (6.8) $\boldsymbol{z}_k^{\mathsf{T}} = \begin{bmatrix} (y_{\mathrm{r,soll},k+1} - y_{\mathrm{r,soll},k})\,\Delta t & y_{\mathrm{r,soll},k} \end{bmatrix}$.

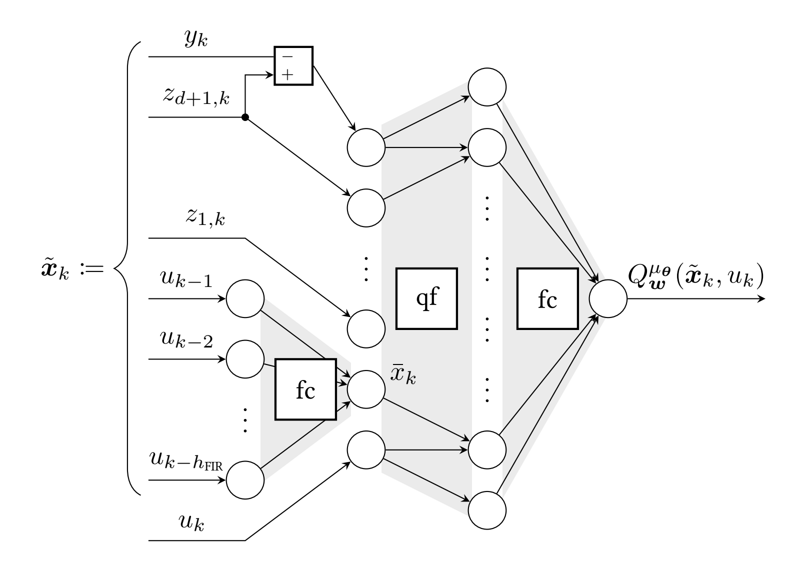


Abbildung 6.3: Verwendete Netzwerkarchitektur für die Approximation der Q-Function. Hierbei ist d der Grad des Polynoms der Sollgeschwindigkeitsapproximation mithilfe der Basisfunktionen $\boldsymbol{\rho}(\kappa)$ und des Gewichts $\boldsymbol{z}_k$. In der ersten Schicht werden einerseits der Geschwindigkeitsfehler $y_{\mathrm{r}}(\boldsymbol{z}_k) - y_k = z_{d+1,k} - y_k$, der Parameter $\boldsymbol{z}_k$ und die aktuelle Stellgröße u_k direkt übergeben. Andererseits wird mithilfe einer vollständig verbundenen Schicht ((engl.): *fully connected layer*, gekennzeichnet durch ‚fc') gemäß (6.10) aus h_{FIR} vergangenen Stellgrößen eine Schätzung des Hilfszustands $\bar{x}_k$ durchgeführt. Die zweite Schicht (gekennzeichnet durch ‚qf') berechnet jede multiplikative Kombination aus der vorherigen Schicht und entspricht somit quadratischen Basisfunktionen. Die dritte Schicht ist wiederum eine vollständig verbundene Schicht, welche die geschätzte Q-Function ausgibt.

6.1.2.3 Gesamtalgorithmus des ADP-basierten Geschwindigkeitsreglers

Der ADP-basierte Regelungsalgorithmus gliedert sich in die drei Teile *Regelung*, *Datenaufbereitung* und *Training*, die nachfolgend erläutert werden. Dabei erfolgt die *Regelung* mit einer durch Δt gegebenen Abtastzeit. Demgegenüber ist die Updaterate des *Trainings* durch Δl gegeben. Um die Echtzeitanforderung der *Regelung* zu gewährleisten, kann Δl von Δt abweichen. Eine grafische Übersicht über den Gesamtalgorithmus ist in Abbildung 6.4 gegeben. Eine kurze Einführung in während des Trainingsvorgangs verwendete Actor-Critic-Mechanismen ist zudem in Anhang D.2 gegeben.

Regelung

Der Block *Regelung* bildet die Schnittstelle zum realen System. Die Eingangsgrößen sind durch die gemessene Geschwindigkeit y_k und einen von außen vorgegebenen Sollgeschwindigkeitsverlauf $y_{\mathrm{r,soll},k}, \ldots, y_{\mathrm{r,soll},k+n_{\mathrm{h}}-1}$ definiert, wohingegen die Stellgröße u_k die Ausgabe darstellt. Innerhalb dieses Blocks wird einerseits der erweiterte Zustand $\tilde{\boldsymbol{x}}_k$ gebildet und andererseits die Stellgröße u_k berechnet.

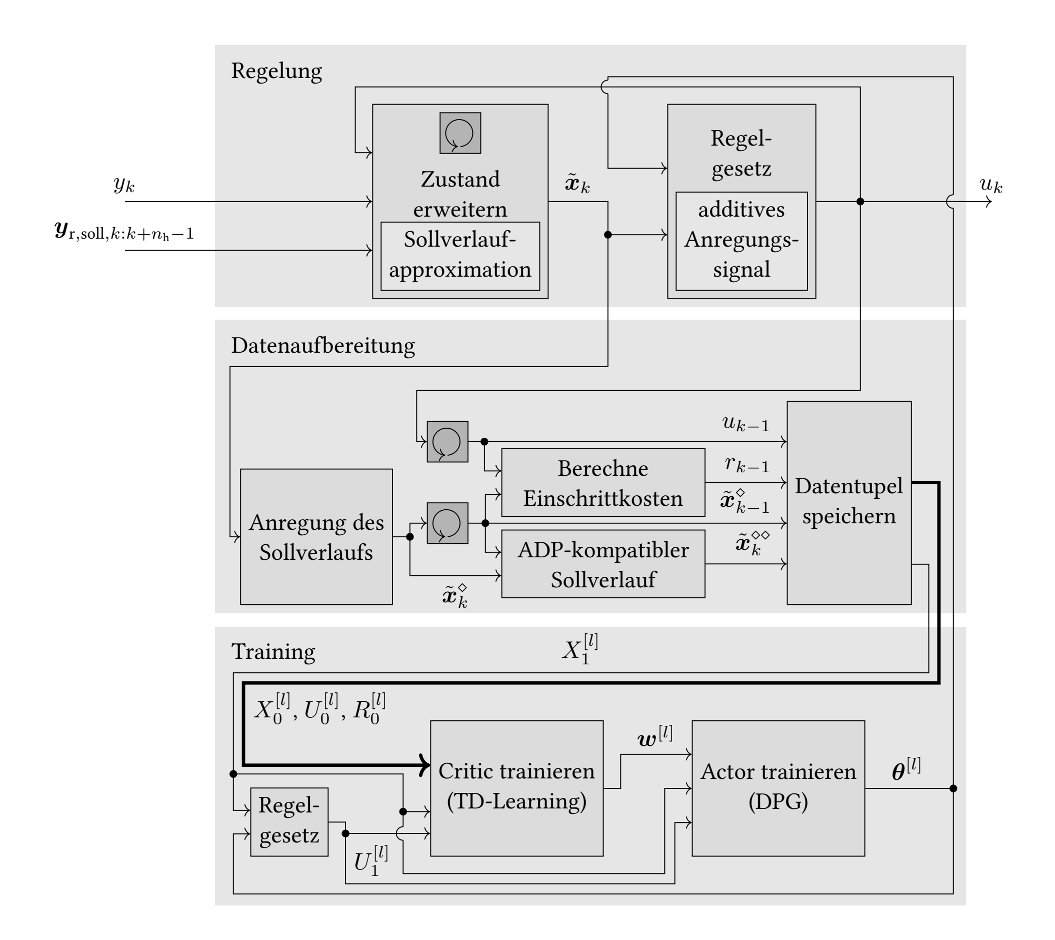


Abbildung 6.4: Übersicht über den Gesamtalgorithmus des ADP-basierten Geschwindigkeitsreglers. Die gezeigte Gesamtstruktur entspricht dem Block *ADP-Regler* in Abbildung 6.2. Der Block *Regelung* übergibt mit der Abtastzeit Δt die Stellgröße u_k an das reale System. Die *Datenaufbereitung* übernimmt Datenvorverarbeitungsschritte, um ADP-kompatible Datentupel zu erzeugen und zu speichern. Im Block *Training* werden diese Datentupel schließlich zum Critic- und Actor-Training genutzt. Dieser Trainingsschritt findet mit der Updaterate Δl statt.

Für die Zustandserweiterung werden zunächst mithilfe eines Ringspeichers Stellgrößen gespeichert, um den Vektor $\boldsymbol{u}_{k-h_\mathrm{FIR}:k-1}$ vergangener Stellgrößen zu erhalten. Weiterhin wird der Referenzparameter $\boldsymbol{z}_k$, wie in (6.8) beschrieben, berechnet. Der erweiterte Zustand $\tilde{\boldsymbol{x}}_k$ wird schließlich aus der gemessenen Geschwindigkeit y_k, den Parametern $\boldsymbol{z}_k$ und den vergangenen Stellgrößen $\boldsymbol{u}_{k-h_\mathrm{FIR}:k-1}$ gebildet (vgl. Abbildung 6.3).

Basierend auf dem aktuellen Reglergewicht $\boldsymbol{\theta}^{[l]}$ und dem erweiterten Zustand $\tilde{\boldsymbol{x}}_k$ wird anschließend mithilfe des Regelgesetzes $\mu_{\boldsymbol{\theta}^{[l]}}(\tilde{\boldsymbol{x}}_k)$ die Stellgröße berechnet. Dabei ist $\mu_{\boldsymbol{\theta}^{[l]}}(\tilde{\boldsymbol{x}}_k)$

durch eine lineare Abhängigkeit des Regelfehlers $y_\mathrm{r}(\boldsymbol{z}_k) - y_k = z_{d+1,k} - y_k$ und, im Fall $d > 0$, die übrigen Solltrajektorienparameter $z_{d,k}, \ldots, z_{1,k}$ gegeben:

$$\mu_{\boldsymbol{\theta}^{[l]}}(\tilde{\boldsymbol{x}}_k) = \begin{bmatrix}(z_{d+1,k} - y_k) & z_{d,k} & \ldots & z_{1,k}\end{bmatrix}^\mathsf{T} \boldsymbol{\theta}^{[l]}. \tag{6.12}$$

Während des Trainingsvorgangs wird zudem ein Anregungssignal zu $\mu_{\boldsymbol{\theta}^{[l]}}(\tilde{\boldsymbol{x}}_k)$ addiert, um die Stellgröße u_k zu erhalten. Dieses wird in Abschnitt 6.1.3.1 konkretisiert. Letztlich wird der erweiterte Zustand $\tilde{\boldsymbol{x}}_k$ sowie die Stellgröße u_k an den Block *Datenaufbereitung* übergeben.

Datenaufbereitung

Der Block *Datenaufbereitung* übernimmt Datenvorverarbeitungsschritte, um ADP-kompatible Datentupel zu erzeugen, zu speichern, und dem nachfolgenden *Training* zur Verfügung zu stellen. Insbesondere werden hier auch die Referenzparameter $\boldsymbol{z}_k$ modifiziert. Diese Modifikation verfolgt zwei Ziele. Einerseits werden die Referenzparameter angeregt, andererseits wird dafür gesorgt, dass die beim Training verwendeten Datentupel $\left\{X_0^{[l]}, U_0^{[l]}, R_0^{[l]}, X_1^{[l]}\right\}$ ADP-kompatibel sind.

Ersteres wird durch Addition von Zufallszahlen zu den Polynomkoeffizienten $\boldsymbol{z}_k$ erreicht. Dieses Vorgehen ist aufgrund der Off-Policy-Charakteristik zulässig (vgl. Anhang D.2). Im Folgenden wird hierzu eine mittelwertfreie Gaußverteilung mit Standardabweichung 1 verwendet. In Abbildung 6.4 kennzeichnet $\tilde{\boldsymbol{x}}_k^{\diamond}$, dass dieses zusätzliche Anregungssignal zum Referenzparameter $\boldsymbol{z}_k$ addiert wird, der Teil des erweiterten Zustands $\tilde{\boldsymbol{x}}_k$ ist.

Ein Pufferspeicher verzögert den erweiterten Zustand $\tilde{\boldsymbol{x}}_k^{\diamond}$, sodass $\tilde{\boldsymbol{x}}_{k-1}^{\diamond}$ zur Verfügung steht. Die Größe $\tilde{\boldsymbol{x}}_k^{\diamond\diamond}$ stellt schließlich ADP-Kompatibilität sicher. $\tilde{\boldsymbol{x}}_k^{\diamond\diamond}$ entsteht, indem in $\tilde{\boldsymbol{x}}_k^{\diamond}$ der Solltrajektorienparameter $\boldsymbol{z}_k$ durch $\boldsymbol{z}_{k-1}^{(1)}$ ersetzt wird, also durch eine um einen Zeitschritt verschobene Version des durch $\boldsymbol{z}_{k-1}$ beschriebenen Solltrajektorienverlaufs aus $\tilde{\boldsymbol{x}}_{k-1}^{\diamond}$ (vgl. (6.3), (6.5) und Bemerkung 3.2). Mithilfe der vergangenen Stellgröße u_{k-1} und $\tilde{\boldsymbol{x}}_{k-1}^{\diamond}$ werden zudem die Einschrittkosten r_{k-1} berechnet. Das so entstandene Tupel $\{\tilde{\boldsymbol{x}}_{k-1}^{\diamond}, u_{k-1}, r_{k-1}, \tilde{\boldsymbol{x}}_k^{\diamond\diamond}\}$ wird in einem Ringspeicher gespeichert, der M Datentupel fasst. Sobald dieser Ringspeicher gefüllt ist, können Batches $\left\{X_0^{[l]}, U_0^{[l]}, R_0^{[l]}, X_1^{[l]}\right\}$, die aus M_B Datentupeln bestehen, aus diesem Speicher gezogen und dem nachfolgenden Trainingsalgorithmus übergeben werden.

Training

In jedem Trainingsschritt wird zunächst das aktuelle Regelgesetz $\mu_{\boldsymbol{\theta}^{[l]}}$ (vgl. (6.12)) für jedes $\tilde{\boldsymbol{x}}$ in $X_1^{[l]}$ ausgewertet, um $U_1^{[l]}$ zu berechnen. Damit wird der Trainingsbatch schließlich zu $\left\{X_0^{[l]}, U_0^{[l]}, R_0^{[l]}, X_1^{[l]}, U_1^{[l]}\right\}$ vervollständigt. Dieser Batch[124] wird dann genutzt, um das Critic-Gewicht $\boldsymbol{w}$ basierend auf dem quadrierten TD-Fehler (D.2) anzupassen. Hierbei wird

[124] Dieses Vorgehen ist unter dem Begriff *Experience Replay* bekannt (vgl. Anhang D.2).

der Levenberg-Marquardt-Algorithmus [Mor78] mit Norm-Clipping [PMB13] verwendet[125]. Anschließend wird das geschätzte Critic-Gewicht $\boldsymbol{w}^{[l]}$ genutzt, um mithilfe des Deterministic Policy Gradients [SLH+14] (vgl. Anhang D.2) die Reglergewichte $\boldsymbol{\theta}^{[l]}$ zu adaptieren[126]. Anzumerken sei an dieser Stelle, dass der Trainingsschritt nicht direkt von externen Signalen abhängt. Die Updaterate Δl, mit der das Training durchgeführt wird, kann somit unabhängig von der Abtastzeit Δt, mit der die Messung der Geschwindigkeit y_k erfolgt und Stellgrößen u_k auf das System appliziert werden, gewählt sein. Eine geeignete Wahl von Δl hängt hierbei insbesondere von der verfügbaren Rechenkapazität ab.

6.1.3 ADP-Solltrajektorienfolgeregelung im Realfahrzeug

In diesem Abschnitt wird gezeigt, dass der vorgestellte ADP-Sollgeschwindigkeitsfolgeregler online in einem Realfahrzeug lernt, vorgegebenen Geschwindigkeitsprofilen zu folgen. Zunächst wird der experimentelle Aufbau beschrieben. Anschließend wird der Online-Trainingsvorgang im Realfahrzeug betrachtet und anhand von Validierungsfahrten ausgewertet. Dabei wird der vorgestellte ADP-Regler, der den Sollgeschwindigkeitsverlauf durch ein Polynom ersten Grades approximiert, mit einem ADP-Regler verglichen, der eine konstante Sollgeschwindigkeit zugrunde legt.

6.1.3.1 Online-Training im Realfahrzeug

Für das Online-Training wird ein auf einem *BMW 740Li* basierendes Versuchsfahrzeug mit Verbrennungsmotor und Automatikgetriebe genutzt (siehe Abbildung 6.1). Dieses Fahrzeug ist mit einer *dSpace Autobox*[127] mit *DS1007* Prozessorboard[128] ausgestattet, die über das fahrzeuginterne Bussystem Zugriff auf das Bremssystem, den Antriebsstrang und weitere Regler hat. Diese Plattform kann mehrere Aufgaben in Echtzeit ausführen und stellt unter anderem die Schnittstelle zu dem vorgestellten ADP-basierten Geschwindigkeitsregler dar. So werden Geschwindigkeitsmessungen y_k zur Verfügung gestellt, der lokale Sollgeschwindigkeitsverlauf $y_{\mathrm{r,soll},k}, y_{\mathrm{r,soll},k+1}, \ldots, y_{\mathrm{r,soll},k+n_\mathrm{h}-1}$ verarbeitet und die resultierende Stellgröße u_k der

[125] Diese Wahl des Optimierungsalgorithmus hat sich als geeignet erwiesen, auch, da vergleichsweise wenige Hyperparameter angepasst werden müssen [PRH19]. Der Levenberg-Marquardt-Algorithmus stellt eine Trust-Region-Optimierungsmethode dar, die neben derselben Approximation der Hessematrix wie bei der Gauß-Newton-Methode einen zusätzlichen Regularisierungsterm nutzt [NW06, S. 258]. Der Regularisierungsterm wird zu $\lambda_{\mathrm{LM}}\boldsymbol{I}$ gesetzt und mittels einfacher Schrittweitensteuerung angepasst. Bei einer Verbesserung des Optimierungsziels wird $\lambda_{\mathrm{LM}} = \frac{\lambda_{\mathrm{LM}}}{3}$ gesetzt, andernfalls $\lambda_{\mathrm{LM}} = 5\lambda_{\mathrm{LM}}$. Initial gilt $\lambda_{\mathrm{LM}} = 1$. Die Verwendung von Norm Clipping, d. h. die Normierung des Gradienten, sobald dessen Norm einen nutzerdefinierten Schwellwert überschreitet, stellt insbesondere eine Begrenzung der Lernrate des Actor-Gewichts sicher. Da die Adaption des Actor-Gewichts auf einer Schätzung des Critic-Gewichts basiert, sollte die Anpassung des Actor-Gewichts langsamer als die Anpassung des Critic-Gewichts stattfinden.

[126] Genau wie bei der Anpassung des Critic-Gewichts wird auch hier der Levenberg-Marquardt-Algorithmus [Mor78] mit Norm-Clipping [PMB13] verwendet.

[127] Datenblatt verfügbar unter: `https://www.dspace.com/shared/data/pdf/2019/dSPACE_AutoBox_PHS_Catalog2019.pdf` (Zugriff am 19.10.2021).

[128] Datenblatt verfügbar unter: `https://www.dspace.com/shared/data/pdf/2019/dSPACE_DS1007_Catalog2019.pdf` (Zugriff am 19.10.2021).

unterlagerten Regelung übergeben. Die Datenaufzeichnung erfolgt mittels eines Ethernet-Anschlusses durch einen PC.

Die verwendeten Hyperparameter sind Tabelle D.1 in Anhang D.3 zu entnehmen. Die Wahl einer kleinen maximalen Norm für den Adaptionsschritt des Actor-Gewichts $\boldsymbol{\theta}^{[l]}$ führt zwar zu einer vergleichsweise langsamen Adaption, soll jedoch Rauschen im Adaptionsprozess des Actors reduzieren. Dies ist darin begründet, dass die Adaption von $\boldsymbol{\theta}^{[l]}$ gemäß (D.5) von der erfolgreichen Schätzung des Critic-Gewichts $\boldsymbol{w}$ abhängt. Aufgrund der begrenzten Rechenkapazität der verwendeten Hardware wird zudem eine Updaterate des Trainingsvorgangs von $\Delta l = 0{,}6\,\mathrm{s}$ gewählt (vgl. Tabelle D.1), während die Abtastzeit des Reglers mit $\Delta t = 0{,}02\,\mathrm{s}$ der Abtastzeit der Systemausgangsgröße y_k entspricht und somit die gegebenen Echtzeitanforderungen erfüllt.

In beiden Trainingsdurchgängen (Ordnung $d = 1$ bzw. $d = 0$ für die Approximation des Sollgeschwindigkeitsprofils) wird das Fahrzeug im ersten Gang betrieben. Die Longitudinaldynamik, die aus Antriebsstrang, Bremssystem und unterlagerter Regelung besteht, dämpft hohe Frequenzen von u_k und weist daher Tiefpassverhalten auf. Der ADP-Mechanismus, der zum Training des Geschwindigkeitsreglers genutzt wird, erfordert jedoch, wie alle adaptiven Methoden, eine ausreichende Systemanregung, um eine Verbesserung der Critic- und Actor-Gewichte zu erzielen. Da mittelwertfreies, hochfrequentes Rauschen aufgrund des Tiefpassverhaltens, ähnlich wie im Beispiel in Abbildung 5.9 gezeigt, kaum Einfluss auf das System hätte, ist es als Anregungssignal für die Stellgröße ungeeignet. Die Ergebnisse aus Kapitel 5 motivieren daher die Verwendung eines Anregungssignals, das zu einer deutlicheren Beeinflussung des Systems führt. Im konkreten Fall des vorliegenden ADP-basierten Geschwindigkeitsreglers hat sich die Addition eines Zufallssignals zum Reglerausgang, der aus einem initial gegebenen, suboptimalen Regler resultiert, bewährt. Dieses Zufallssignal wird dabei alle $2\,\mathrm{s}$ zufällig aus einer Gleichverteilung im Intervall $[-1\,\mathrm{m\,s^{-2}}, 1\,\mathrm{m\,s^{-2}}]$ gezogen. Während des Trainingsvorgangs wird die Geschwindigkeit zwischen ca. $6\,\mathrm{km\,h^{-1}}$ und $30\,\mathrm{km\,h^{-1}}$ variiert. Ein beispielhafter Ausschnitt der Geschwindigkeit y_k und der Stellgröße u_k während des Trainingsvorgangs ist Abbildung 6.5 zu entnehmen.

Um in den Trainingstupeln $\left\{X_0^{[l]}, U_0^{[l]}, R_0^{[l]}, X_1^{[l]}, U_1^{[l]}\right\}$ auch eine Anregung der Referenztrajektorienparameter $\boldsymbol{z}_k$ zu erreichen, werden, wie in Abschnitt 6.1.2.3 beschrieben, verrauschte Parameter $\boldsymbol{z}_{k-1}$ verwendet. Die Verteilung der während des Trainingsvorgangs genutzten verrauschten Referenzparameter $\boldsymbol{z}_{k-1}$, aus denen mithilfe von $\boldsymbol{D}(1)$ die ADP-kompatiblen Parameter $\boldsymbol{z}_{k-1}^{(1)}$ erzeugt werden, ist für $d = 1$ in Abbildung 6.6 gezeigt[129].

Abbildung 6.7 zeigt den Verlauf der Reglergewichte $\boldsymbol{\theta}^{[l]}$ für die beiden Durchgänge mit linearer Referenzapproximation (Ordnung $d = 1$) und konstanter Referenzapproximation (Ordnung $d = 0$) während des Trainingsvorgangs. Die Verstärkung, die den konstanten Anteil der Sollgeschwindigkeitsapproximation beschreibt, verhält sich dabei für beide Fälle ähnlich. Nach etwa $1300\,\mathrm{s}$ verbleiben die Reglerparameter in einem kleinen Intervall.

129 Für den Fall $d = 0$ entfällt der durch z_1 gegebene lineare Anteil.

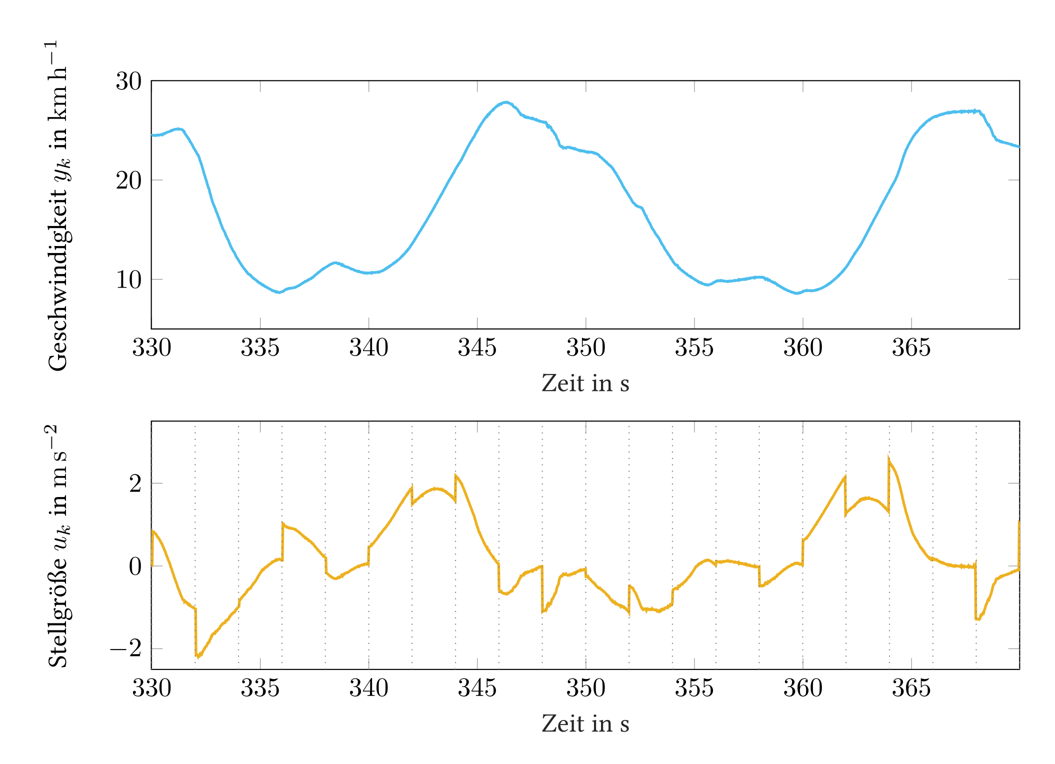


Abbildung 6.5: Beispielhafter Ausschnitt der Geschwindigkeit y_k sowie der Stellgröße u_k während des Trainingsvorgangs. Im Sinne der Systemanregung wird alle 2 s ein Zufallswert aus einer Gleichverteilung auf dem Intervall $[-1\,\mathrm{m\,s^{-1}}, 1\,\mathrm{m\,s^{-1}}]$ gezogen, der zum Reglerausgang addiert wird. Die Zeitpunkte, zu denen dieser Zufallswert gezogen wird, sind durch vertikale, gepunktete Linien gekennzeichnet.

6.1.3.2 Auswertung der gelernten Regler im Realfahrzeug

Abschließend sollen die beiden gelernten Regler anhand einer Validierungsfahrt verglichen werden. Dazu werden die Reglergewichte $\boldsymbol{\theta}^{[l]}$ am Ende der beiden Trainingsdurchgänge konstant gehalten. Zudem wird das Anregungssignal entfernt und eine Auswertungsfahrt durchgeführt.

In Abbildung 6.8 ist ersichtlich, dass mit beiden Reglern die gewünschte Sollgeschwindigkeit erreicht wird. Aufgrund des zu minimierenden Gütefunktionals (6.2), das unter anderem den Stellaufwand bestraft[130], sowie der begrenzten Vorausschaufähigkeit der Regler, wird dem vorgegebenen Geschwindigkeitsprofil zwar nicht exakt gefolgt, aufgrund der prädiktiven Eigenschaften des Reglers mit linearer Sollgeschwindigkeitsapproximation ist für $d = 1$ der zeitliche Versatz von y_k bezüglich der Sollgeschwindigkeit im Vergleich zum Regler mit $d = 0$ jedoch merklich reduziert.

[130] Hierdurch werden insbesondere im Sinne des Fahrkomforts hohe Beschleunigungen vermieden (vgl. [DLL18]).

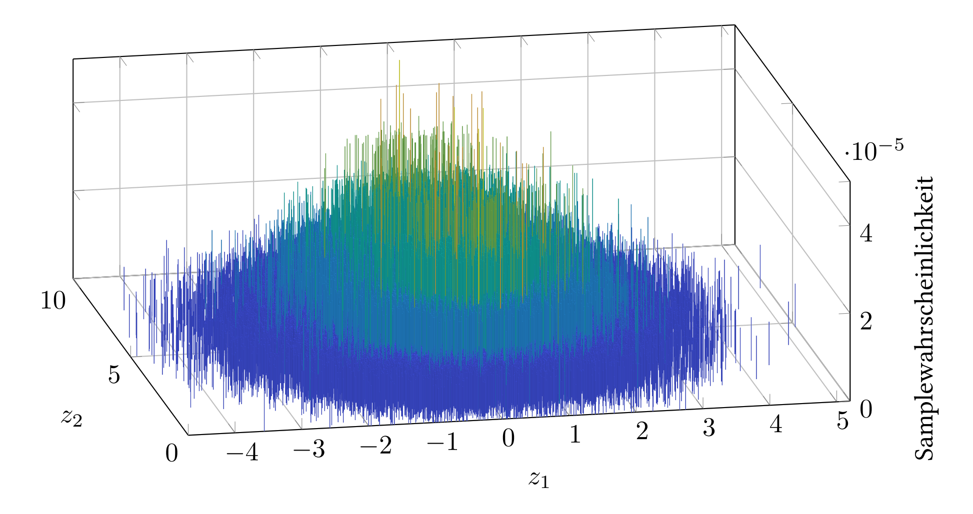


Abbildung 6.6: Verteilung der während des Trainingsvorgangs verwendeten verrauschten Referenzparameter $\boldsymbol{z}_{k-1}$. Dabei stellt z_2 die Sollgeschwindigkeit (konstanter Anteil in $\mathrm{m\,s^{-1}}$) und z_1 die Steigung der Sollgeschwindigkeit (linearer Anteil in $\mathrm{m\,s^{-2}}$) dar.

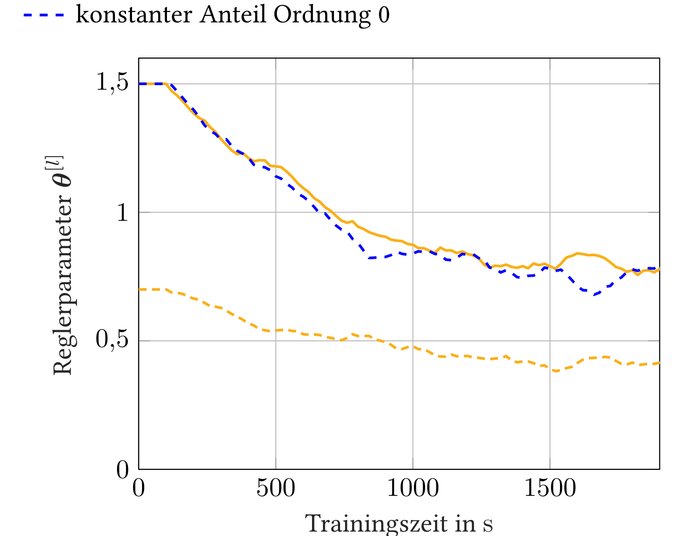


Abbildung 6.7: Reglerparameter $\boldsymbol{\theta}^{[l]}$ während des Trainingsvorgangs. Während der ersten 100 s wird zunächst der Speicher der Größe M mit Datentupeln gefüllt.

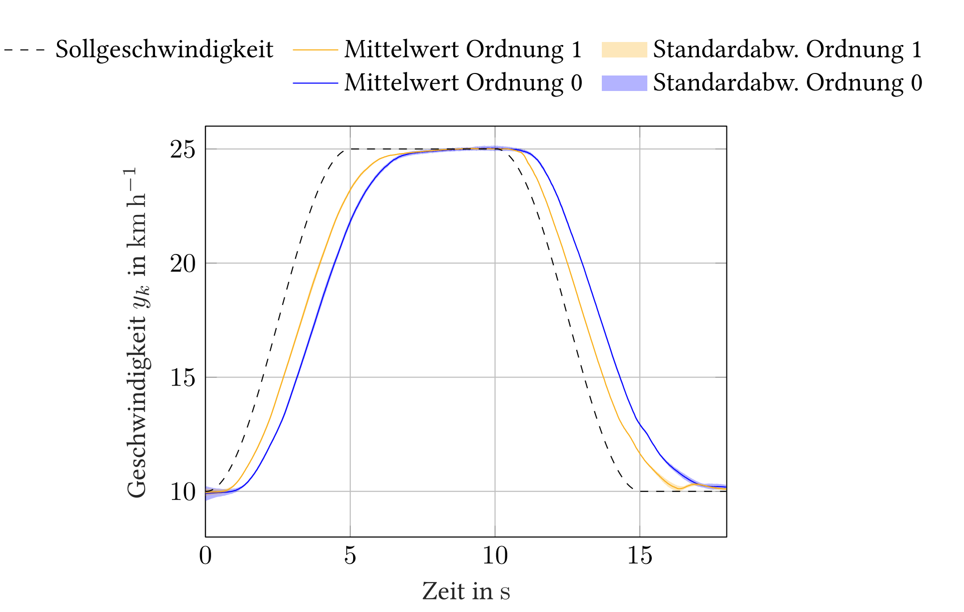


Abbildung 6.8: Sollgeschwindigkeitsverlauf und gemessene Geschwindigkeit y_k während der Auswertungsfahrt nach dem Training. Als beispielhaftes Sollgeschwindigkeitsprofil wurde ein um 17,5 km h^{-1} verschobener Sinusverlauf mit einer Amplitude von 7,5 km h^{-1} und einer Periodendauer von 10 s, der an den Extrempunkten für jeweils 5 s konstant gehalten wird, gewählt. Die Ergebnisse sind über 20 Durchgänge gemittelt. Der Regler, der eine lineare Approximation des Sollgeschwindigkeitsverlaufs berücksichtigt ($d = 1$), weist dabei eine geringere Abweichung vom Sollverlauf auf, als der Vergleichsregler mit konstanter Geschwindigkeit ($d = 0$).

Die aktuelle Sollgeschwindigkeitsabweichung, die durch den konstanten Anteil des Polynoms nullten Grades ($d = 0$) bzw. des Polynoms ersten Grades ($d = 1$) berücksichtigt wird, hat bei beiden Reglern einen nahezu identischen Einfluss. Dies ist den konstanten Anteilen der Solltrajektorienparameter $\boldsymbol{z}_k$ in Abbildung 6.9 sowie den konstanten Anteilen der Reglerparameter $\boldsymbol{\theta}^{[l]}$ in Abbildung 6.7 zu entnehmen. Der Regler mit linearer Approximation des Sollgeschwindigkeitsprofils kann jedoch vorausschauender handeln, da der lineare Anteil der Referenzapproximation als Vorsteuerterm agiert (vgl. (6.12) und [Lun20a, S. 11]). Diese zusätzliche Stellgröße hängt dabei direkt mit der aktuellen Steigung der Sollgeschwindigkeit zusammen, d. h. der aktuellen Sollbeschleunigung, die dem linearen Anteil des Solltrajektorienparameters $\boldsymbol{z}_k$ in Abbildung 6.9 entspricht. Aufgrund dieses zusätzlichen Vorsteuerterms sendet der Regler, der eine lineare Approximation des Sollverlaufs nutzt, die durch u_k gegebenen Beschleunigungssignale früher als der Vergleichsregler. Dies ist in Abbildung 6.10 gezeigt.

Abschließend werden noch die mit den beiden Reglern während der Auswertungsfahrt resultierenden Kosten im Sinne des Gütefunktionals (6.2) betrachtet. Da dieses Gütemaß im

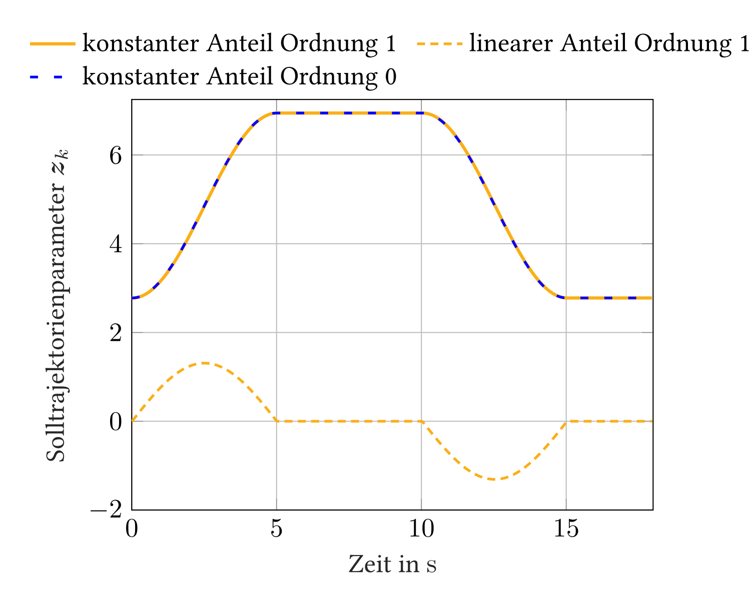


Abbildung 6.9: Solltrajektorienparameter $\boldsymbol{z}_k$ während der Auswertungsfahrt. Der konstante Anteil repräsentiert die aktuelle Sollgeschwindigkeit, der lineare Anteil die aktuelle Sollbeschleunigung.

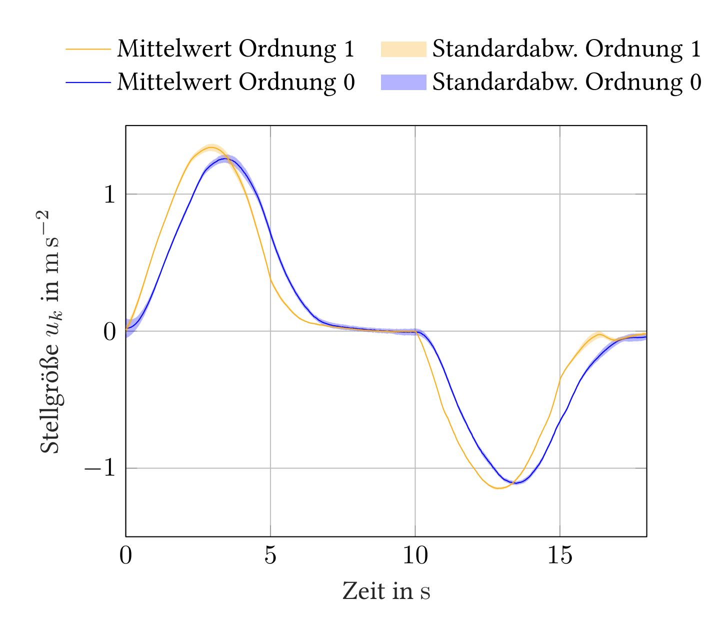


Abbildung 6.10: Stellgröße u_k während des Auswertungsmanövers (gemittelt über 20 Durchgänge). Die Amplituden der Stellgröße sind für die betrachteten Fälle vergleichbar, jedoch reagiert der Regler mit $d = 1$ früher und handelt vorausschauender.

Realversuch aufgrund des unendlichen Zeithorizonts in (6.2) nicht berechnet werden kann, wird als Näherung

$$V^{\mu}\left(y_k, \boldsymbol{z}_k\right) \approx \sum_{\kappa=0}^{100} \gamma^{\kappa} r\left(y_{k+\kappa}, y_{\mathrm{r}}\left(\boldsymbol{z}_k^{(\kappa)}\right), \mu\left(y_{k+\kappa}, \boldsymbol{z}_k^{(\kappa)}\right)\right) \tag{6.13}$$

genutzt. Diese Näherung, die aufgrund der verwendeten Diskontierung $\gamma = 0{,}95$ (vgl. Tabelle D.1) zulässig ist, da Kosten in ferner Zukunft nur wenig zur Value Function V^{μ} beitragen, gibt einen Eindruck über die Güte der beiden Regler. Wie in Abbildung 6.11 ersichtlich ist, reduziert der vorgestellte Regler, der eine lineare Approximation des Sollgeschwindigkeitsverlaufs verwendet, die Kosten (6.13) deutlich gegenüber dem Vergleichsregler. Die besseren Vorausschaueigenschaften und der daraus resultierende verringerte zeitliche Versatz führen zu geringeren Regelabweichungen und damit letztlich zu geringeren Kosten.

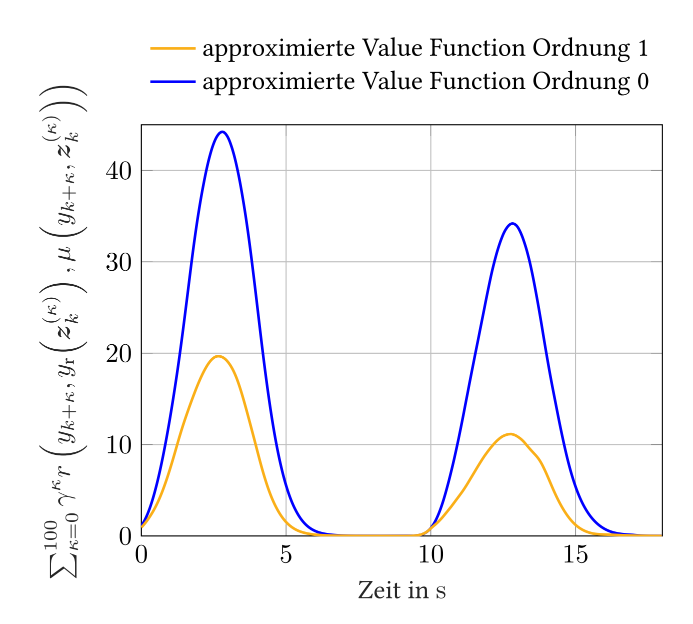


Abbildung 6.11: Approximierte Value Function während des Auswertungsmanövers. Der Regler, der eine lineare Approximation des Sollverlaufs verwendet ($d = 1$), führt dabei zu deutlich geringeren Kosten als der Vergleichsregler.

6.1.4 Diskussion

In diesem Abschnitt wurde, basierend auf der in Abschnitt 3.2 vorgestellten ADP-kompatiblen Referenztrajektoriendarstellung, ein lernender Geschwindigkeitsregler für ein Realfahrzeug vorgestellt, der online, d. h. während der Trainingsfahrt, die Schätzparameter der Q-Function und die Reglerparameter adaptiert. Dieser Ansatz zeichnet sich durch die explizite Einbeziehung der lokalen linearen Approximation des Sollgeschwindigkeitsverlaufs aus. Die Konsistenz

der Trainingsdaten mit der Bellman-Gleichung wird hierbei durch die Definition eines erweiterten Zustands $\tilde{\boldsymbol{x}}_k$ sowie der virtuellen Verschiebung der Solltrajektorienparameter $\boldsymbol{z}_k$ erreicht.

Das erfolgreiche Online-Training in einem Realfahrzeug demonstriert die grundsätzliche Anwendbarkeit der in dieser Arbeit entwickelten, neuartigen, ADP-basierten Solltrajektorienregler in realen regelungstechnischen Problemstellungen und liefert somit einen anwendungsorientierten Beitrag zu Forschungsfrage 1 aus Abschnitt 2.4.1. Insbesondere offenbart die explizite Einbeziehung des approximierten Solltrajektorienverlaufs in die Value Function Vorteile. So führt die zusätzliche Verwendung der Änderungsrate der aktuellen Sollgeschwindigkeit zu besseren Vorausschaueigenschaften und signifikant reduzierten Kosten.

Eine effizientere Implementierung der verwendeten Algorithmen sowie ein auf einer externen Recheneinheit durchgeführter Trainingsprozess können künftig die Updaterate Δl des Trainings wesentlich reduzieren und ein schnelleres Training ermöglichen. Ein schnellerer Trainingsvorgang erleichtert schließlich mögliche Erweiterungen, zum Beispiel das Training in anderen Geschwindigkeitsbereichen und Gängen oder die Einbeziehung zusätzlicher Daten, wie beispielsweise das aktuelle Straßenprofil.

Im nächsten Abschnitt wird die in dieser Arbeit vorgestellte ADP-kompatible Referenztrajektoriendarstellung auf ein zweites reales regelungstechnisches Problem angewandt und ein daraus resultierender modellfreier ADP-Regler mit einem modellbasierten Regler verglichen.

6.2 Modellfreier Trajektorienfolgeregler für ein reales Ball-auf-Platte-System

In diesem Abschnitt wird die in Kapitel 3 vorgestellte ADP-kompatible parametrierte Referenztrajektoriendarstellung verwendet, um einen Solltrajektorienfolgeregler für ein reales Ball-auf-Platte-System, das in Abbildung 6.12 gezeigt ist, zu trainieren[131]. Obwohl Ball-auf-Platte-Systeme weitverbreitete Beispielsysteme in der Regelungstechnik darstellen, sind die in der Literatur vorhandenen Regler hierfür entweder vollständig modellbasiert [KIB+19], [ABB+02], [BFGB12], [DHS17], [KCS03] oder modellbasiert mit zusätzlicher Fuzzy-Regelung [MSV08]. Im Gegensatz zu bestehenden Reglern benötigt der im Folgenden verwendete Regelungsansatz kein exaktes Modell des Ball-auf-Platte-Systems, da Messdaten des realen Systems dazu verwendet werden, einen optimierungsbasierten Solltrajektorienfolgeregler zu trainieren. Dieses Vorgehen vermeidet daher eine Modellbildung, die beispielsweise durch konstruktive Eigenschaften, wie einer vertikal versetzten Drehachse der Platte, erschwert wird (vgl. [KIB+19]). Ebenso wird eine anschließende manuelle Feinjustage nicht mehr benötigt. Somit stellt die vorliegende Arbeit die erste Anwendung eines modellfreien ADP-basierten Solltrajektorienfolgeregelungsansatzes auf ein reales Ball-auf-Platte-System dar.

[131] Ergebnisse dieses Abschnitts wurden im Rahmen eines Konferenzbeitrags veröffentlicht [KKIH21].



Abbildung 6.12: Ball-auf-Platte-System des Instituts für Regelungs- und Steuerungssysteme am Karlsruher Institut für Technologie, das für die modellfreie, ADP-basierte Solltrajektorienfolgeregelung verwendet wird.

Weiterhin erlauben existierende Regler für Ball-auf-Platte-Systeme bisher entweder überhaupt keine Solltrajektorienvorgabe der Ballposition [KIB+19], [ABB+02], [BFGB12] oder betrachten nur die aktuelle Abweichung der Ballposition von einer stationären Sollpositionsvorgabe [DHS17], [KCS03], [MSV08]. Letzteres führt jedoch zu einer merklichen Zeitverzögerung im Vergleich zum gewünschten Trajektorienverlauf. Demgegenüber wird in der vorliegenden Arbeit eine ADP-kompatible lokale Approximation des Solltrajektorienverlaufs der Ballposition explizit in den Regler integriert. Diese flexible, aber dennoch kompakte Darstellung ermöglicht prädiktives Verhalten des resultierenden Reglers.

Die Nutzung eines Off-Policy-Ansatzes (vgl. Abschnitt 2.1.4.4) erlaubt zudem die Wiederverwendung der Messdaten und somit eine dateneffiziente Implementierung. Die Verwendung einer On-Policy-Methode würde nicht nur dazu führen, dass zwingend benötigtes Anregungsrauschen (vgl. Kapitel 5) zu einem Offset in der Schätzung der Critic-Gewichte führen könnte,

sondern insbesondere nach jedem Policy-Improvement-Schritt komplett neue Messdaten vom realen System aufgezeichnet werden müssten. Der vorgestellte ADP-basierte Solltrajektorienfolgeregler mit lokaler Approximation des Sollpositionsverlaufs wird schließlich sowohl mit einem ADP-basierten Regler mit stationärer Sollvorgabe als auch mit modellbasierten Reglern, die ebenfalls eine lokale Approximation des Sollpositionsverlaufs bzw. eine konstante Sollvorgabe verwenden, verglichen.

6.2.1 Ball-auf-Platte-System und Problembeschreibung

Nachfolgend wird zunächst das verwendete reale Ball-auf-Platte-System, das ohne genaue Kenntnis eines Systemmodells mithilfe eines ADP-Solltrajektorienfolgeregelungsansatzes geregelt werden soll, vorgestellt und anschließend die Problemstellung definiert.

6.2.1.1 Ball-auf-Platte-System

Das in dieser Arbeit verwendete und in Abbildung 6.12 gezeigte Ball-auf-Platte-System besitzt eine quadratische Platte mit einer Seitenlänge von $1\,\mathrm{m}$ und einer Masse von $16{,}3\,\mathrm{kg}$. Die Platte kann in zwei zueinander orthogonalen Richtungen, die im Folgenden durch X und Y gekennzeichnet sind, geneigt werden. Für beide Plattendimensionen steht hierfür ein eigener Motor zur Verfügung. Die Plattenwinkel $(\alpha^{[X]}, \alpha^{[Y]})$ und zugehörigen Winkelgeschwindigkeiten $(\omega^{[X]}, \omega^{[Y]})$ werden alle $10\,\mathrm{ms}$ gemessen. Ein Ball mit einer Masse von $0{,}042\,\mathrm{kg}$ und einem Radius von $0{,}02\,\mathrm{m}$ befindet sich auf der Platte. Seine Position in einem plattenfesten Koordinatensystem wird kamerabasiert erfasst, sodass alle $\Delta t = 40\,\mathrm{ms}$ die Ballposition $(s^{[X]}, s^{[Y]})$ sowie die Ballgeschwindigkeit $(v^{[X]}, v^{[Y]})$ zur Verfügung stehen. Für $\iota \in \mathcal{D} = \{X, Y\}$ werden die Systemzustände der jeweiligen Plattendimension zu

$$\boldsymbol{x}_k^{[\iota]} = \left[s_k^{[\iota]} \; v_k^{[\iota]} \; \alpha_k^{[\iota]} \; \omega_k^{[\iota]} \right]^\mathsf{T} \tag{6.14}$$

zusammengefasst. Die Eingangsgrößen des Systems sind durch die Motorströme $u_k^{[\iota]} = I_k^{[\iota]}$ gegeben. Die Systemarchitektur ist in Abbildung 6.13 skizziert. Eine detailliertere Beschreibung der Systemarchitektur und der verwendeten Hardware ist in [KIB+19], [Kil20] und [Bla18] zu finden[132].

Da die beiden Plattendimensionen X und Y sich nur geringfügig gegenseitig beeinflussen, ist eine getrennte Regelung in beiden Dimensionen üblich [ABB+02], [BFGB12], [KCS03], [KIB+19]. Die Dynamik des Ball-auf-Platte-Systems ist zwar in X- und Y-Richtung unterschiedlich, jedoch werden die Regler für beide Plattendimensionen grundsätzlich mit demselben Verfahren trainiert. Daher wird auf den Index ι im Folgenden aus Gründen der Lesbarkeit zumeist verzichtet.

[132] Im Gegensatz zu [KIB+19] und [Bla18] wird in der vorliegenden Arbeit eine schwerere Platte sowie ein anderer Ball verwendet. Dies wurde insbesondere bei der Auslegung des modellbasierten Vergleichsreglers berücksichtigt.

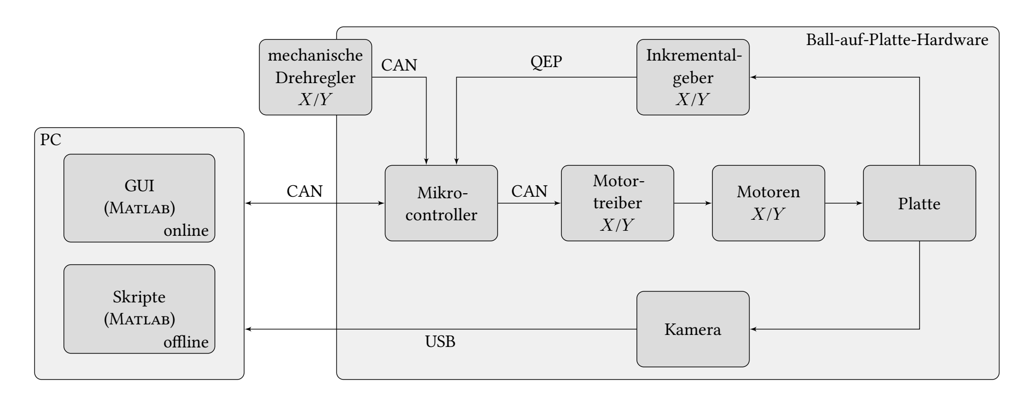


Abbildung 6.13: Architektur des Ball-auf-Platte-Systems. Abbildung nach [Kil20].

6.2.1.2 Problemstellung

Betrachtet werde ein zeitdiskretes, steuerbares System mit der Dynamik

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right), \tag{6.15}$$

wobei $k \in \mathbb{N}_{\geq 0}$ den diskreten Zeitschritt indiziert, $\boldsymbol{x}_k \in \mathbb{R}^n$ dem Systemzustand entspricht (vgl. (6.14)), $\boldsymbol{u}_k \in R^p$ die Stellgröße I_k darstellt, und $\boldsymbol{F}$ unbekannt ist. Aus Abschnitt 6.2.1.1 folgt für jede Dimension in $\mathcal{D}$ die Systemordnung $n = 4$ sowie eine Stellgrößenanzahl von $p = 1$. In jedem Zeitschritt k sei eine lokale Approximation des Solltrajektorienverlaufs der Ballposition durch

$$s_{\mathrm{r}}\left(\boldsymbol{z}_k^{(\kappa)}\right) := s_{\mathrm{r}}(\boldsymbol{z}_k, \kappa) = \boldsymbol{z}_k^{\intercal}\boldsymbol{\rho}(\kappa) \tag{6.16}$$

mit $\kappa \in \mathbb{N}_{\geq 0}$ gegeben, wobei $s_{\mathrm{r}}\left(\boldsymbol{z}_k^{(\kappa)}\right)$ die Sollposition im Zeitschritt $k + \kappa$ bezeichnet. Der Vektor $\boldsymbol{z}_k \in \mathbb{R}^{d+1}$ parametriert den Sollverlauf und $\boldsymbol{\rho}(\kappa)$ bezeichnet gegebene Basisfunktionen (vgl. (3.15)).

Die folgende Problemstellung formalisiert das Ziel, dass die Ballposition einem vorgegebenen Verlauf folgen soll, während für die Werte der übrigen Systemzustände bei entsprechender Parametrierung eine möglichst geringe Abweichung von null gefordert werden kann. Außerdem soll der Stellaufwand gering gehalten werden.

Problem 6.1

Gegeben seien Basisfunktionen $\boldsymbol{\rho}(\kappa)$ zur ADP-kompatiblen Solltrajektorienapproximation nach Definition 3.1 sowie M Messtupel $\{\boldsymbol{x}_\kappa, u_\kappa, \boldsymbol{x}_{\kappa+1}\}$, $\kappa = k, \ldots, k+M-1$. Die Systemdynamik $\boldsymbol{F}(\boldsymbol{x}_k, u_k)$ sei unbekannt. Gesucht ist das Regelgesetz $\mu^*(\boldsymbol{x}_k, \boldsymbol{z}_k)$, sodass $\forall \boldsymbol{x}_k, \boldsymbol{z}_k$ die durch $u_k^* = \mu^*(\boldsymbol{x}_k, \boldsymbol{z}_k)$ gegebene Stellgröße das Gütefunktional

$$\begin{aligned} J_k &:= \sum_{\kappa=0}^{\infty} \gamma^\kappa \left(\begin{bmatrix} x_{1,k+\kappa} - s_\mathrm{r}\left(\boldsymbol{z}_k^{(\kappa)}\right) \\ x_{2,k+\kappa} \\ x_{3,k+\kappa} \\ x_{4,k+\kappa} \end{bmatrix}^\mathsf{T} \boldsymbol{Q} \begin{bmatrix} x_{1,k+\kappa} - s_\mathrm{r}\left(\boldsymbol{z}_k^{(\kappa)}\right) \\ x_{2,k+\kappa} \\ x_{3,k+\kappa} \\ x_{4,k+\kappa} \end{bmatrix} + u_{k+\kappa}^\mathsf{T} R u_{k+\kappa} \right) \\ &=: \sum_{\kappa=0}^{\infty} \gamma^\kappa r\left(\boldsymbol{x}_{k+\kappa}, u_{k+\kappa}, s_\mathrm{r}\left(\boldsymbol{z}_k^{(\kappa)}\right)\right) \end{aligned} \tag{6.17}$$

minimiert. Hierbei sei $\gamma \in (0, 1]$ ein Diskontierungsfaktor, $\boldsymbol{Q}$ positiv semidefinit und R positiv definit.

6.2.2 ADP-Solltrajektorienfolgeregler für ein Ball-auf-Platte-System

Für die optimale Q-Function

$$\begin{aligned} Q^*(\boldsymbol{x}_k, \boldsymbol{z}_k, u_k) &= r\big(\boldsymbol{x}_k, s_\mathrm{r}(\boldsymbol{z}_k), u_k\big) + \sum_{\kappa=1}^{\infty} \gamma^\kappa r\left(\boldsymbol{x}_{k+\kappa}, s_\mathrm{r}\left(\boldsymbol{z}_k^{(\kappa)}\right), \mu^*\left(\boldsymbol{x}_{k+\kappa}, \boldsymbol{z}_k^{(\kappa)}\right)\right) \\ &= r\big(\boldsymbol{x}_k, s_\mathrm{r}(\boldsymbol{z}_k), u_k\big) + \gamma Q^*\left(\boldsymbol{x}_{k+1}, \boldsymbol{z}_k^{(1)}, \mu^*\left(\boldsymbol{x}_{k+1}, \boldsymbol{z}_k^{(1)}\right)\right) \end{aligned} \tag{6.18}$$

mit $\boldsymbol{z}_k^{(\kappa)}$ nach Definition 3.2 folgt gemäß Lemma 3.1, dass

$$u_k^* = \arg\min_{u_k} Q^*(\boldsymbol{x}_k, \boldsymbol{z}_k, u_k) \tag{6.19}$$

eine Lösung für Problem 6.1 darstellt. Da die optimale Q-Function $Q^*(\boldsymbol{x}_k, \boldsymbol{z}_k, u_k)$ a priori unbekannt ist, sei $\hat{Q}^*(\boldsymbol{x}_k, \boldsymbol{z}_k, u_k) = \hat{\boldsymbol{w}}^\mathsf{T} \boldsymbol{\phi}(\boldsymbol{x}_k, \boldsymbol{z}_k, u_k)$, mit dem zu schätzenden Gewicht $\hat{\boldsymbol{w}} \in \mathbb{R}^h$ und $\boldsymbol{\phi}(\cdot) \in \mathbb{R}^h$ (vgl. (3.24)). Mithilfe einer Policy Iteration (vgl. Abschnitt 2.1.4.1) wird in der l-ten Iteration im Policy-Evaluation-Schritt die geschätzte Q-Function

$$\hat{Q}^{[l+1]}(\boldsymbol{x}_k, \boldsymbol{z}_k, u_k) = \hat{\boldsymbol{w}}^{[l+1]\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{x}_k, \boldsymbol{z}_k, u_k), \tag{6.20}$$

welche die Gesamtkosten des aktuellen Regelgesetzes $\hat{\mu}^{[l]}$ beschreibt, gesucht. Hierzu wird $\hat{\boldsymbol{w}}^{[l+1]}$ adaptiert, um nach Möglichkeit die Gleichung

$$\hat{Q}^{[l+1]}(\boldsymbol{x}_k, \boldsymbol{z}_k, u_k) = r\big(\boldsymbol{x}_k, s_\mathrm{r}(\boldsymbol{z}_k), u_k\big) + \gamma \hat{Q}^{[l+1]}\left(\boldsymbol{x}_{k+1}, \boldsymbol{z}_k^{(1)}, \hat{\mu}^{[l]}\left(\boldsymbol{x}_{k+1}, \boldsymbol{z}_k^{(1)}\right)\right) \tag{6.21}$$

zu erfüllen. Der anschließende Policy-Improvement-Schritt ist dann durch

$$\hat{\mu}^{[l+1]}(\boldsymbol{x}_k, \boldsymbol{z}_k) = \arg\min_{u_k} \hat{Q}^{[l+1]}(\boldsymbol{x}_k, \boldsymbol{z}_k, u_k) \tag{6.22}$$

gegeben. In den nächsten Abschnitten wird die verwendete lokale Solltrajektorienapproximation, die Approximation der Q-Function und der Trainingsalgorithmus beschrieben.

6.2.2.1 Quadratische lokale Referenzapproximation

Im Folgenden werde der Solltrajektorienverlauf der Ballposition aus lokaler Perspektive zum Zeitschritt k durch ein quadratisches Polynom ($d = 2$)

$$s_{\mathrm{r}}\left(\boldsymbol{z}_k^{(\kappa)}\right) = \boldsymbol{z}_k^{\mathsf{T}} \boldsymbol{\rho}(\kappa) = p_{k,2}(\kappa\Delta t)^2 + p_{k,1}\kappa\Delta t + p_{k,0} \tag{6.23}$$

mit $\boldsymbol{\rho}(\kappa) := \begin{bmatrix}(\kappa\Delta t)^2 & \kappa\Delta t & 1\end{bmatrix}^{\mathsf{T}}$ und dem Parametervektor $\boldsymbol{z}_k := \begin{bmatrix}p_{k,2} & p_{k,1} & p_{k,0}\end{bmatrix}^{\mathsf{T}}$ approximiert. Die Abtastzeit ist durch Δt gegeben. Die Verschiebungsmatrix $\boldsymbol{D}(\kappa)$, die benötigt wird, um den propagierten Parameter $\boldsymbol{z}_k^{(\kappa)}$ nach Definition 3.2 aus $\boldsymbol{z}_k$ zu berechnen, ergibt sich aus

$$\begin{aligned}
s_{\mathrm{r}}\left(\boldsymbol{z}_k^{(\kappa+j)}\right) &= s_{\mathrm{r}}\left(\boldsymbol{z}_k^{(\kappa)}, j\right) \\
&= \boldsymbol{z}_k^{\mathsf{T}} \boldsymbol{\rho}(\kappa + j) \\
&= \boldsymbol{z}_k^{\mathsf{T}} \begin{bmatrix}\left((\kappa+j)\Delta t\right)^2 \\ (\kappa+j)\Delta t \\ 1\end{bmatrix} \\
&= \boldsymbol{z}_k^{\mathsf{T}} \underbrace{\begin{bmatrix}1 & 2\kappa\Delta t & (\kappa\Delta t)^2 \\ 0 & 1 & \kappa\Delta t \\ 0 & 0 & 1\end{bmatrix}}_{=:\boldsymbol{D}(\kappa)} \boldsymbol{\rho}(j) \\
&= \boldsymbol{z}_k^{(\kappa)\mathsf{T}} \boldsymbol{\rho}(j),
\end{aligned} \tag{6.24}$$

$\forall \kappa, j \in \mathbb{N}_{\geq 0}$. Für einen beliebigen Verlauf der Sollposition $s_{\mathrm{r,soll},k}$ ist in jedem Zeitschritt k ein Parametervektor $\boldsymbol{z}_k$ gesucht, sodass $s_{\mathrm{r}}\left(\boldsymbol{z}_k^{(\kappa)}\right)$, $\kappa \in \mathbb{N}_{\geq 0}$, eine Approximation von $s_{\mathrm{r,soll},k+\kappa}$ darstellt. Während der Laufzeit wird dabei angenommen, dass der Sollpositionsverlauf über einen Vorausschauhorizont von $n_{\mathrm{h}} \in \mathbb{N}_{>0}$ Zeitschritten vorliegt. In jedem Zeitschritt k wird $\boldsymbol{z}_k$ dann mittels gewichteter Least-Squares-Regression geschätzt. Analog zu (3.71) und (3.72) folgt

$$\boldsymbol{z}_k^{\mathsf{T}} = s_{\mathrm{r,soll},k:k+n_{\mathrm{h}}-1} \boldsymbol{W}_{\mathrm{p}} \boldsymbol{\rho}_{0,n_{\mathrm{h}}-1} \left(\boldsymbol{\rho}_{0,n_{\mathrm{h}}-1}^{\mathsf{T}} \boldsymbol{W}_{\mathrm{p}} \boldsymbol{\rho}_{0,n_{\mathrm{h}}-1}\right)^{-1} \tag{6.25}$$

mit

$$s_{\mathrm{r,soll},k:k+n_{\mathrm{h}}-1} := \begin{bmatrix}s_{\mathrm{r,soll},k} & s_{\mathrm{r,soll},k+1} & \dots & s_{\mathrm{r,soll},k+n_{\mathrm{h}}-1}\end{bmatrix}, \tag{6.26}$$

$$\boldsymbol{W}_{\mathrm{p}} := \operatorname{diag}(1, \beta, \dots, \beta^{n_{\mathrm{h}}-1}), \quad \beta \leq 1, \tag{6.27}$$

$$\boldsymbol{\rho}_{0:n_{\mathrm{h}}-1} := \begin{bmatrix}\boldsymbol{\rho}(0) & \boldsymbol{\rho}(1) & \dots & \boldsymbol{\rho}(n_{\mathrm{h}}-1)\end{bmatrix}^{\mathsf{T}}. \tag{6.28}$$

6.2.2.2 Approximation der Q-Function und Policy Iteration

Die Approximation der Q-Function nach (6.20) wird zu

$$\begin{aligned}\hat{Q}^{[l]}(\boldsymbol{x}_k, \boldsymbol{z}_k, u_k) &:= \begin{bmatrix} \boldsymbol{x}_k \\ u_k \\ \boldsymbol{z}_k \\ 1 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \boldsymbol{h}_{\mathrm{xx}}^{[l]} & \boldsymbol{h}_{\mathrm{xu}}^{[l]} & \boldsymbol{h}_{\mathrm{xz}}^{[l]} & h_{\mathrm{x1}}^{[l]} \\ \boldsymbol{h}_{\mathrm{ux}}^{[l]} & h_{\mathrm{uu}}^{[l]} & \boldsymbol{h}_{\mathrm{uz}}^{[l]} & h_{\mathrm{u1}}^{[l]} \\ \boldsymbol{h}_{\mathrm{zx}}^{[l]} & \boldsymbol{h}_{\mathrm{zu}}^{[l]} & \boldsymbol{h}_{\mathrm{zz}}^{[l]} & h_{\mathrm{z1}}^{[l]} \\ h_{\mathrm{1x}}^{[l]} & h_{\mathrm{1u}}^{[l]} & h_{\mathrm{1z}}^{[l]} & h_{\mathrm{11}}^{[l]} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ u_k \\ \boldsymbol{z}_k \\ 1 \end{bmatrix} \\ &=: \bar{\bar{\boldsymbol{y}}}_k^{\mathsf{T}} \boldsymbol{H}^{[l]} \bar{\bar{\boldsymbol{y}}}_k =: \hat{\boldsymbol{w}}^{[l]\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{x}_k, \boldsymbol{z}_k, u_k) \end{aligned} \tag{6.29}$$

(vgl. (3.46)) mit $\boldsymbol{H}^{[l]} = \boldsymbol{H}^{[l]\mathsf{T}}$ gewählt. Somit gilt $\boldsymbol{\phi}(\boldsymbol{x}_k, \boldsymbol{z}_k, \boldsymbol{u}_k) = \bar{\bar{\boldsymbol{y}}}_k \otimes_{\mathrm{r}} \bar{\bar{\boldsymbol{y}}}_k$ und $\hat{\boldsymbol{w}}^{[l]}$ entspricht den nicht-redundanten Elementen der symmetrischen Matrix $\boldsymbol{H}^{[l]}$[133]. Für jede Dimension $\iota \in \mathcal{D}$ sind im Fall einer lokalen Referenzapproximation nach Abschnitt 6.2.2.1 mit $d = 2$ somit $h = 45$ Gewichte zu lernen. Die Wahl einer quadratischen Q-Function ist durch die in [KIB$^+$19] veröffentlichte Regelung des betrachteten Ball-auf-Platte-Systems mithilfe eines modellbasierten linear-quadratischen Optimalregelungsansatzes motiviert. Nach Lemma 3.2 weist die Q-Function eines LQ-Problems eine quadratische Struktur auf.

Im Policy-Evaluation-Schritt zur Erfüllung von (6.21) wird der Least-Squares-Temporal-Difference-Q-Learning-Ansatz [LP03] mit der in [LP03, Abschnitt 5.2] beschriebenen Fixpunktbedingung verwendet[134]. Somit werden M Tupel $\left\{\boldsymbol{x}_k, u_k, \boldsymbol{x}_{k+1}, \boldsymbol{z}_k, \boldsymbol{z}_k^{(1)}\right\}$ genutzt, um

$$\hat{\boldsymbol{w}}^{[l+1]} = \left(\bar{\boldsymbol{\Phi}}^{\mathsf{T}} \left(\bar{\boldsymbol{\Phi}} - \gamma \bar{\boldsymbol{\Phi}}^{+[l]}\right)\right)^{-1} \bar{\boldsymbol{\Phi}}^{\mathsf{T}} \boldsymbol{r} \tag{6.30a}$$

zu bestimmen. Dabei gilt

$$\bar{\boldsymbol{\Phi}} := \begin{bmatrix} \boldsymbol{\phi}_1^{\mathsf{T}} \\ \vdots \\ \boldsymbol{\phi}_M^{\mathsf{T}} \end{bmatrix}, \quad \bar{\boldsymbol{\Phi}}^{+[l]} := \begin{bmatrix} \boldsymbol{\phi}_1^{+[l]\mathsf{T}} \\ \vdots \\ \boldsymbol{\phi}_M^{+[l]\mathsf{T}} \end{bmatrix}, \quad \boldsymbol{r} := \begin{bmatrix} r_1 \\ \vdots \\ r_M \end{bmatrix} \tag{6.30b}$$

---

133 Elemente von $\hat{\boldsymbol{w}}^{[l]}$, die zu Nebendiagonalelementen von $\boldsymbol{H}^{[l]}$ gehören, werden dabei jeweils mit dem Faktor 2 multipliziert, um die redundanten Elemente der symmetrischen Matrix $\boldsymbol{H}^{[l]}$ zu berücksichtigen.

134 Der Bellman-Operator $\mathcal{B}Q(\boldsymbol{x}_k, \boldsymbol{z}_k, u_k) := r\big(\boldsymbol{x}_k, s_{\mathrm{r}}(\boldsymbol{z}_k), u_k\big) + \gamma Q\left(\boldsymbol{x}_{k+1}, \boldsymbol{z}_k^{(1)}, \hat{\mu}^{[l]}\left(\boldsymbol{x}_{k+1}, \boldsymbol{z}_k^{(1)}\right)\right)$ stellt nach [BBdE10, S. 24] eine Kontraktion mit dem Kontraktionsfaktor γ in der L_∞-Norm dar, deren Fixpunkt durch die korrekte Q-Function zum betrachteten Regelgesetz $\hat{\boldsymbol{\mu}}^{[l]}(\cdot)$ gegeben ist. Da die Anwendung des Bellman-Operators und die Forderung $\hat{Q}^{[l+1]} \overset{!}{=} \mathcal{B}\hat{Q}^{[l+1]}$ jedoch zu einer geschätzten Q-Function führen kann, die möglicherweise nicht mehr in dem durch $\boldsymbol{\phi}(\cdot)$ aufgespannten Raum liegt, wird eine orthogonale Projektion mit der Projektionsmatrix $\bar{\boldsymbol{\Phi}}\left(\bar{\boldsymbol{\Phi}}^{\mathsf{T}}\bar{\boldsymbol{\Phi}}\right)^{-1}\bar{\boldsymbol{\Phi}}^{\mathsf{T}}$ [Mey00, S. 430] mit $\bar{\boldsymbol{\Phi}}$ nach (6.30b) vorgenommen. Dies führt zur Forderung $\hat{Q}^{[l+1]} \overset{!}{=} \bar{\boldsymbol{\Phi}}\left(\bar{\boldsymbol{\Phi}}^{\mathsf{T}}\bar{\boldsymbol{\Phi}}\right)^{-1}\bar{\boldsymbol{\Phi}}^{\mathsf{T}}\mathcal{B}\hat{Q}^{[l+1]}$ (vgl. [LP03, S. 1117]). Eine ausführliche Beschreibung ist in [LP03], [BBdE10] und [Kil20, Abschnitt 2.3.6] zu finden. Für den Fall, dass die Q-Function exakt durch die gewählten Funktionsapproximatoren abgebildet werden kann, führt die verwendete Fixpunktforderung zur selben Lösung wie die in Abschnitt 3.2.3 beschriebene Methode (vgl. [LP03, Abschnitt 5.3]). Da die Fixpunktforderung nach Aussage von [LP03, Abschnitt 5.3] jedoch in Experimenten häufig zu besseren Regelgesetzen führt und für die Anwendung empfohlen wird, findet diese Methode im Folgenden Verwendung.

mit

$$\boldsymbol{\phi}_k \coloneqq \boldsymbol{\phi}\left(\boldsymbol{x}_k, \boldsymbol{z}_k, u_k\right), \quad k = 1, \ldots, M, \tag{6.30c}$$

$$r_k \coloneqq r\big(\boldsymbol{x}_k, s_{\mathrm{r}}(\boldsymbol{z}_k), u_k\big) \tag{6.30d}$$

und

$$\boldsymbol{\phi}_k^{+[l]} \coloneqq \boldsymbol{\phi}\left(\boldsymbol{x}_{k+1}, \boldsymbol{z}_k^{(1)}, \hat{\mu}^{[l]}\left(\boldsymbol{x}_{k+1}, \boldsymbol{z}_k^{(1)}\right)\right). \tag{6.30e}$$

Aufgrund der Off-Policy-Charakteristik können die Messdaten in jeder Iteration wiederverwendet werden. Zudem erfordert die Minimierung in (6.22), dass

$$\frac{\partial \hat{Q}^{[l+1]}}{\partial u_k} = 2\left(\boldsymbol{h}_{\mathrm{ux}}^{[l+1]}\boldsymbol{x}_k + \boldsymbol{h}_{\mathrm{uz}}^{[l+1]}\boldsymbol{z}_k + h_{\mathrm{u1}}^{[l+1]} + h_{\mathrm{uu}}^{[l+1]}u_k\right) \overset{!}{=} \boldsymbol{0} \tag{6.31}$$

gilt. Hieraus folgt für den Policy-Improvement-Schritt (6.22) der analytische Ausdruck

$$\hat{\mu}^{[l+1]}(\boldsymbol{x}_k, \boldsymbol{z}_k) = \underbrace{-\left(h_{\mathrm{uu}}^{[l+1]}\right)^{-1}\left[\boldsymbol{h}_{\mathrm{ux}}^{[l+1]}\ \boldsymbol{h}_{\mathrm{uz}}^{[l+1]}\ h_{\mathrm{u1}}^{[l+1]}\right]}_{=:\boldsymbol{K}^{[l+1]}} \begin{bmatrix}\boldsymbol{x}_k \\ \boldsymbol{z}_k \\ 1\end{bmatrix} \tag{6.32}$$

(vgl. Satz 3.2). Somit hängt der Motorstrom I_k von $\boldsymbol{x}_k, \boldsymbol{z}_k$ und einer statischen Offsetkorrektur ab.

> **Bemerkung 6.3**
> *Die Wahl von $\hat{Q}(\cdot)$ in (6.29) erweitert die in Abschnitt 3.2.4 präsentierte Approximationsstruktur um einen konstanten Term in $\bar{\bar{\boldsymbol{y}}}_k$. Hierdurch wird insbesondere auch die Darstellung eines statischen Offsetstroms ermöglicht. Ein solcher statischer Offsetstrom kann beispielsweise eine asymmetrische Masseverteilung der Platte kompensieren. Während dieser Offsetstrom in [KIB+ 19] heuristisch ermittelt wurde, wird er in der vorliegenden Arbeit somit automatisiert gelernt.*

6.2.2.3 Trainingsablauf

Der zur Adaption eingesetzte LSPI-Algorithmus [LP03] verwendet Trainingsdaten, um iterativ die geschätzte Q-Function und somit das Regelgesetz anzupassen. Die während des Trainings genutzten Datentupel bestehen aus zwei Teilen. Einerseits werden am realen System aufgezeichnete Messdaten verwendet, andererseits werden ADP-kompatible Solltrajektoriendaten erzeugt. Diese Trainingsdaten werden anschließend vorverarbeitet und normiert, bevor eine Policy Iteration zum Training durchgeführt wird.

Reale Messdaten

Während der Aufzeichnung von Messdaten am realen System muss sowohl das System ausreichend angeregt werden[135], als auch ein sicherer Betrieb gewährleistet sein. Inspiriert durch Satz 5.2 und die Simulationsergebnisse aus Abschnitt 5.7.3 hat sich zur Anregung auch im vorliegenden Anwendungsbeispiel eine Überlagerung hinreichend vieler unterschiedlicher Frequenzen, die sich nicht gegenseitig auslöschen, bewährt. Da für die Betrachtung des modellfreien ADP-basierten Ansatzes angenommen wird, dass kein genaues Modell der Systemdynamik vorliegt, ist der Entwurf eines Reglers, der das System geeignet anregt (vgl. beispielsweise Abschnitt 5.7.2) und dabei gleichzeitig einen sicheren Betrieb des Systems gewährleistet, nicht trivial. Daher wird zur Aufzeichnung von Trainingsdaten am realen System eine Anregung des Systems durch einen Menschen genutzt. Dieser kann mithilfe der in Abbildung 6.13 gezeigten mechanischen Drehregler den Plattenwinkel manuell in X- und Y-Richtung beeinflussen. Die am Ball-auf-Platte-System eingebauten Inkrementalgeber sowie eine über der Platte befindliche Kamera zeichnen währenddessen, wie in [KIB$^+$19] beschrieben, die Systemzustände $\boldsymbol{x}_k^{[X]}$ und $\boldsymbol{x}_k^{[Y]}$ nach (6.14) auf. Zudem wird der applizierte Motorstrom gespeichert. Somit werden Datentupel $\{\boldsymbol{x}_k, u_k, \boldsymbol{x}_{k+1}\}$ mit einer Abtastzeit von $\Delta t = 40\,\mathrm{ms}$ erzeugt. An dieser Stelle sei darauf hingewiesen, dass der Mensch bei der Datenaufzeichnung zwar eine ausreichende Anregung sicherstellen muss, also Plattenwinkel und Ballposition variiert werden müssen, jedoch dank der Off-Policy-Charakteristik der anschließend verwendeten ADP-Methode dabei weder optimal noch entsprechend des Initialgewichts $\hat{\boldsymbol{w}}^{[l]}$ oder einer vorgegebenen Solltrajektorie gehandelt werden muss.

ADP-kompatible Solltrajektorien

Wiederum durch die in Kapitel 5 betrachtete Überlagerung mehrerer Anregungsfrequenzen motiviert, wird der Sollpositionsverlauf des Balls für die Trainingsdaten in Form einer Summe aus Sinus- und Kosinusfunktionen konstruiert. Konkret wurde hierzu $s_{\mathrm{r,soll},k}$ durch Abtastung von

$$0{,}1\left(\sin(0{,}4t) + \sin(t) + 0{,}5\sin(1{,}4t) + \cos(1{,}6t) + 0{,}2\sin(2t)\right) \tag{6.33}$$

mit $\Delta t = 40\,\mathrm{ms}$ erzeugt. Zu jedem Zeitschritt k wird dieser Sollverlauf mittels gewichteter Least-Squares-Approximation nach (6.25) durch ein quadratisches Polynom ($d = 2$) mit $\beta = 0{,}8$ und $n_\mathrm{h} = 10$ in Form des Parameters $\boldsymbol{z}_k$ beschrieben. Anschließend wird $\boldsymbol{z}_k$ mithilfe von $\boldsymbol{D}(1)$ nach (6.24) propagiert, um $\boldsymbol{z}_k^{(1)}$ im Sinne einer ADP-kompatiblen Solltrajektoriendarstellung zu erhalten.

Vorverarbeitung und Normierung

Die aufgezeichneten Messdaten $\boldsymbol{x}_k, u_k,$ und $\boldsymbol{x}_{k+1}$ werden zunächst mithilfe eines gleitenden Mittelwertfilters der Länge 5 geglättet. Die resultierenden Zustands- und Stellgrößenverläufe sind Abbildung 6.14 zu entnehmen. Anschließend werden diese Messdaten zusammen mit

[135] Ähnlich wie zuvor in Kapitel 5 und der Rangbedingung in (3.29) muss auch hier eine von den Systemzuständen, Solltrajektorienparametern und Basisfunktionen abhängige Matrix vollen Rang aufweisen. Konkret muss die Existenz der in (6.30a) auftretenden Inverse sichergestellt sein.

den Trainingsreferenzparametern $\boldsymbol{z}_k$ und $\boldsymbol{z}_k^{(1)}$ zu Tupeln $\left\{\boldsymbol{x}_k, u_k, \boldsymbol{x}_{k+1}, \boldsymbol{z}_k, \boldsymbol{z}_k^{(1)}\right\}$ zusammengefasst. Für den Trainingsvorgang werden $M = 1200$ Datentupel verwendet. Bei einer Abtastzeit von $\Delta t = 40\,\mathrm{ms}$ entspricht dies einer Aufzeichnungsdauer von $48\,\mathrm{s}$. Zudem werden die Zustände und Solltrajektorienparameter mit einem Normierungsfaktor $c_{\mathrm{norm}} = 10$ skaliert ($\boldsymbol{x}_{\mathrm{norm},k} = c_{\mathrm{norm}}\boldsymbol{x}_k$ und $\boldsymbol{z}_{\mathrm{norm},k} = c_{\mathrm{norm}}\boldsymbol{z}_k$)[136].

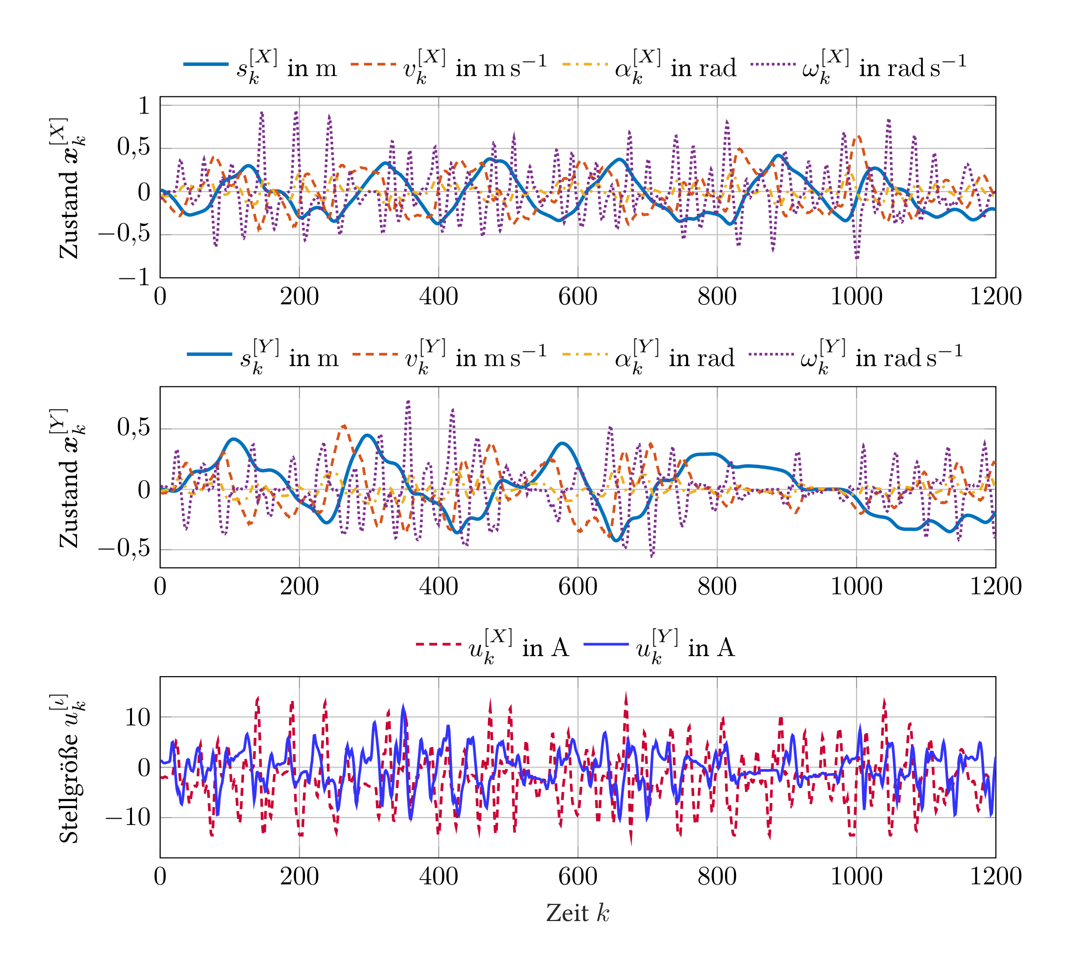


Abbildung 6.14: Mithilfe der mechanischen Drehregler durch einen Menschen aufgezeichnete, geglättete Messdaten des Ball-auf-Platte-Systems. Die Abtastzeit beträgt $\Delta t = 40\,\mathrm{ms}$.

[136] Abhängig vom gewählten Verhältnis der Parameter $\boldsymbol{Q}$ und R im Gütefunktional nach (6.17) ist der Wertebereich der Stellgröße u_k zumeist deutlich größer als der Wertebereich der Zustandsgrößen $\boldsymbol{x}_k$. Die Verwendung des heuristisch ermittelten Skalierungsfaktors c_{norm} wurde daher zugunsten einer verbesserten numerischen Stabilität eingeführt.

Parametrierung und Algorithmus
Primäres Ziel des Reglers ist, dass die Ballposition dem Solltrajektorienverlauf möglichst präzise folgt. Zusätzlich soll die Platte nach Möglichkeit in einer horizontalen Lage gehalten und der Stellaufwand bestraft werden. Im Folgenden werden dazu

$$\boldsymbol{Q}_{\text{norm}} := \frac{\boldsymbol{Q}}{c_{\text{norm}}^2} = \operatorname{diag}(800, 0, 400, 0) \quad \text{und} \quad R = 1 \tag{6.34}$$

gewählt. Hierbei stellt $\boldsymbol{Q}$ die Gewichtungsmatrix nach (6.17) dar und $\boldsymbol{Q}_{\text{norm}}$ entspricht der Gewichtung der normierten Größen $\boldsymbol{x}_{\text{norm},k}$ und $\boldsymbol{z}_{\text{norm},k}$. Somit wird insbesondere eine Abweichung der Ballposition s_k von dem durch $\boldsymbol{z}_k$ parametrierten Sollpositionsverlauf $s_{\text{r}}\left(\boldsymbol{z}_k^{(\kappa)}\right)$ nach (6.23) sowie die Abweichung des Plattenwinkels α_k von der Horizontalen $\alpha = 0$ bestraft. Zudem werde der Diskontierungsfaktor zu $\gamma = 0{,}9$ gewählt. Für die erste Iteration werden alle Gewichte $\hat{\boldsymbol{w}}^{[0]}$ zu 1 initialisiert[137].

Mithilfe der in Abschnitt 6.2.2.2 beschriebenen Q-Function-Approximation sowie des LSPI-Algorithmus wird in jeder Iteration l eine Adaption des Gewichtsvektors $\hat{\boldsymbol{w}}^{[l]}$ vorgenommen[138]. Der Algorithmus endet, sobald die zu

$$\left\| \hat{\boldsymbol{w}}^{[l]} - \hat{\boldsymbol{w}}^{[l-1]} \right\|_2 \leq e_{\hat{\boldsymbol{w}}} = 10^{-6} \tag{6.35}$$

gewählte Abbruchbedingung erfüllt ist. Aus dem letzten Policy-Improvement-Schritt resultiert nach einer Rücknormierung schließlich die Reglermatrix $\boldsymbol{K}$ (vgl. (6.32)) und somit das Regelgesetz

$$\hat{\boldsymbol{\mu}}(\boldsymbol{x}_k, \boldsymbol{z}_k) = -\begin{bmatrix} \boldsymbol{K}_{\text{x}} & \boldsymbol{K}_{\text{z}} & K_{\text{off}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{z}_k \\ 1 \end{bmatrix}. \tag{6.36}$$

Diese Schritte sind im Ablaufdiagramm in Abbildung 6.15 veranschaulicht.

6.2.3 Ergebnisse

Der modellfreie, ADP-basierte Regler $\boldsymbol{K}_{\text{ADP}}$ wird zu Vergleichszwecken mit einem modellbasierten Optimalregler $\boldsymbol{K}_{\text{Modell}}$ verglichen. Da die Regler für die beiden Plattendimensionen X und Y auf gleiche Weise trainiert bzw. berechnet werden, wird zunächst nur eine Dimension betrachtet, bevor schließlich in Abschnitt 6.2.3.4 eine gleichzeitige Regelung beider Plattendimensionen erfolgt.

---

[137] Zwar entspricht dies einem instabilen initialen Regelgesetz, aufgrund der Diskontierung $\gamma = 0,9$ konvergiert der erste Policy-Evaluation-Schritt in diesem Fall aber dennoch gegen eine endliche Lösung und die Policy Iteration letztlich gegen einen stabilisierenden Regler. Unter Nutzung von Vorwissen über die Systemdynamik könnte $\hat{\boldsymbol{w}}^{[0]}$ alternativ so initialisiert werden, dass $\boldsymbol{K}_{\text{x}} = \begin{bmatrix} 10 & 10 & 100 & 10 \end{bmatrix}$ (vgl. (6.36)) gilt. Dies entspricht einem stabilisierenden, jedoch suboptimalen initialen Regelgesetz.

[138] Die Komplexität jeder Iteration wird hierbei durch den Policy-Evaluation-Schritt mit $\mathcal{O}(h^3 + h^2 M)$ dominiert.

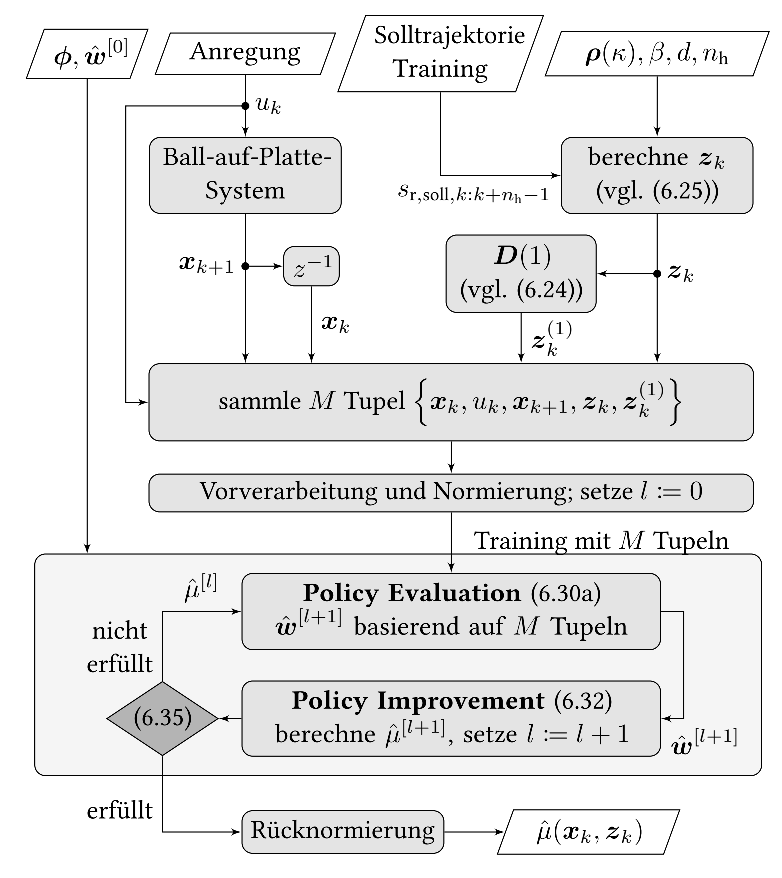


Abbildung 6.15: Ablaufschema des Trainingsvorgangs für den modellfreien ADP-Solltrajektorienfolgeregler für das Ball-auf-Platte-System. Hieraus resultiert der geschätzte Optimalregler $\hat{\mu}(\boldsymbol{x}_k, \boldsymbol{z}_k)$, welcher die durch $\boldsymbol{z}_k$ beschriebene lokale Approximation des Solltrajektorienverlaufs explizit berücksichtigt.

Der ADP-basierte Regler wird wie in Abschnitt 6.2.2.3 beschrieben trainiert. Die Konvergenz von $\hat{\boldsymbol{w}}^{[l]}$ ist in Abbildung 6.16 zu sehen. Die gelernte Reglermatrix ergibt sich schließlich zu

$$\boldsymbol{K}_{\mathrm{ADP}}^{[Y]} = [\underbrace{64{,}8 \quad 32{,}3 \quad 145{,}3 \quad 16{,}2}_{\boldsymbol{K}_\mathrm{x}} \quad \underbrace{-27{,}9 \quad -36{,}9 \quad -60{,}7}_{\boldsymbol{K}_\mathrm{z}} \quad \underbrace{-0{,}1}_{K_\mathrm{off}}]. \tag{6.37}$$

Die modellbasierte Vergleichslösung, die das in [KIB+19] gegebene Systemmodell mit entsprechend angepassten Parametern der verwendeten Platte und des verwendeten Balls nutzt, wird nach Satz 3.1 berechnet. Daraus ergibt sich die Reglermatrix

$$\boldsymbol{K}_{\mathrm{Modell}}^{[Y]} = [\underbrace{75{,}5 \quad 55{,}9 \quad 213{,}3 \quad 33{,}0}_{\boldsymbol{K}_\mathrm{x}} \quad \underbrace{-41{,}1 \quad -55{,}9 \quad -75{,}1}_{\boldsymbol{K}_\mathrm{z}}] \tag{6.38}$$

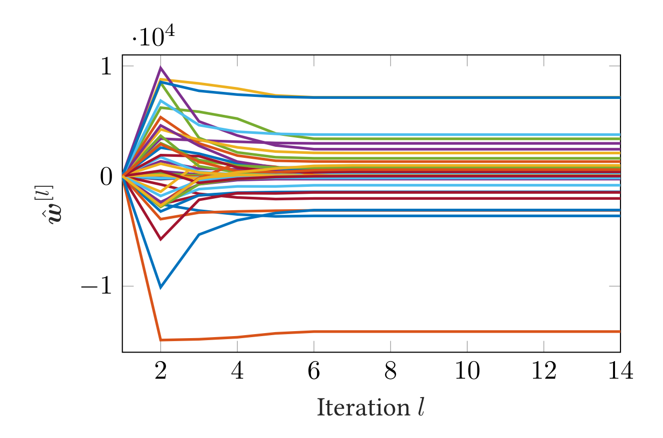


Abbildung 6.16: Gewichtsvektor $\hat{\boldsymbol{w}}^{[l]}$ über die Iterationen l des LSPI-Algorithmus bis zur Erfüllung der Abbruchbedingung nach (6.35).

des modellbasierten Vergleichsreglers. Im Folgenden wird zunächst eine konstante Sollpositionsvorgabe, d. h. $d = 0$, und anschließend die Verwendung einer Approximation des Solltrajektorienverlaufs für Polynome vom Grad $d = 2$ betrachtet.

6.2.3.1 Konstante Sollvorgabe: geglätteter Rechteckverlauf

Der gelernte, modellfreie ADP-Regler wird anhand eines geglätteten Rechteckverlaufs[139] als Sollpositionsvorgabe $s_{\mathrm{r,soll}}$ des Balls mit dem berechneten, modellbasierten Regler verglichen. Hierzu wird zunächst eine konstante Sollvorgabe ($d = 0$) verwendet. In Abbildung 6.17 ist die mittlere Ballposition über 11 Wiederholungen für den gelernten Regler in Blau und für den berechneten, modellbasierten Regler in Gelb gegeben. Die jeweilige Standardabweichung ist transparent dargestellt. Der gesamte Systemzustand $\boldsymbol{x}_k$ sowie die Stellgröße u_k sind in Abbildung D.1 in Anhang D.4 visualisiert.

Bei beiden Reglern liegt eine merkliche zeitliche Verzögerung der Ballposition gegenüber der Sollposition vor. Dies ist darin begründet, dass den Reglern lediglich die aktuelle Sollposition im Zeitschritt k, nicht jedoch deren zukünftiger Verlauf zur Verfügung steht, das System träge ist und die Stellgrößen im Gütefunktional bestraft werden. Der gelernte Regler reagiert dabei etwas schneller und regelt die Ballposition präziser, was sich, wie Abbildung 6.17 zu entnehmen ist, auch in geringeren akkumulierten Einschrittkosten

$$\sum_{\kappa=0}^{k} r\big(\boldsymbol{x}_\kappa, u_\kappa, s_{\mathrm{r}}(\boldsymbol{z}_\kappa)\big) \tag{6.39}$$

widerspiegelt.

[139] Genauer gesagt wird der Sollpositionsverlauf aus einem um 0,15 m verschobenen sinusförmigen Signal mit einer Amplitude von 0,15 m und einer Periodendauer von 147 Zeitschritten, das im Maximum für 200 Zeitschritte konstant gehalten wird, erzeugt.

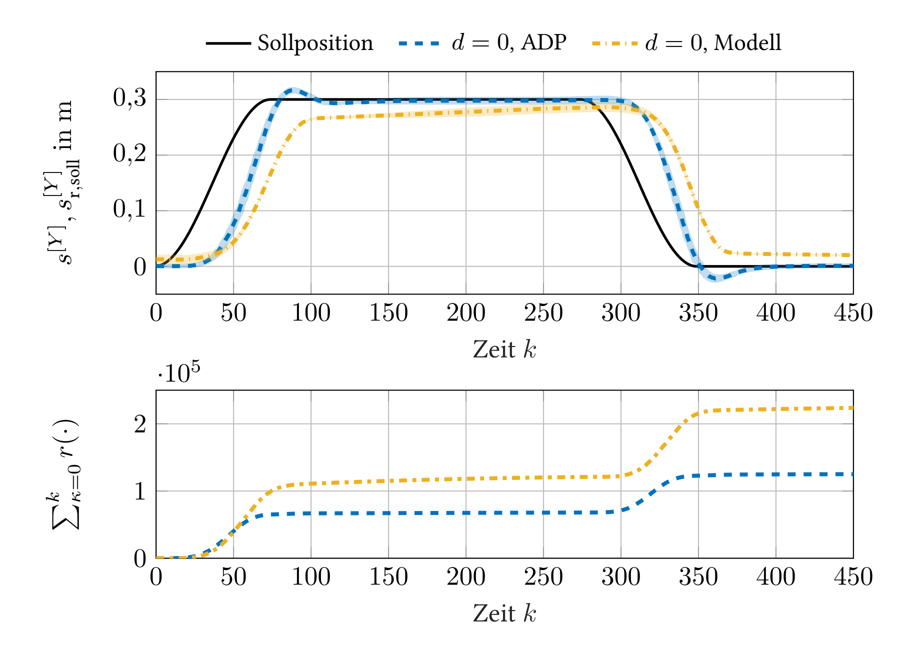


Abbildung 6.17: Vorgabe einer stationären Sollposition ($d = 0$) für den gelernten, ADP-basierten Regler (blau) sowie den modellbasierten Vergleichsregler (gelb). *Oben:* Mittlere Ballposition und Standardabweichung über 11 Wiederholungen. *Unten:* Mittlere akkumulierte Einschrittkosten.

6.2.3.2 Approximation des Sollverlaufs: geglätteter Rechteckverlauf

Für $d = 2$ zeigt Abbildung 6.18 den Vergleich eines gelernten, ADP-basierten Solltrajektorienfolgereglers (rot) mit einem modellbasierten Solltrajektorienfolgeregler (grün). Beide Regler folgen dem Sollpositionsverlauf deutlich besser als der Regler mit konstanter Sollvorgabe, der zu Vergleichszwecken in Blau dargestellt ist[140]. Aufgrund der lokalen Approximation der Solltrajektorie nutzen die Regler mit $d = 2$ Information über den zukünftigen Verlauf der Sollposition. Dies ermöglicht signifikant geringere akkumulierte Einschrittkosten, wie Abbildung 6.18 zu entnehmen ist. Ähnlich wie bei den in Abschnitt 6.2.3.1 betrachteten Reglern mit konstanter Sollvorgabe weist auch im Fall $d = 2$ der gelernte, ADP-basierte Regler geringere Gesamtkosten im Vergleich zum modellbasierten Regler auf.

6.2.3.3 Approximation des Sollverlaufs: Validierungstrajektorie

In diesem Abschnitt wird die Vorgabe eines Sollpositionsverlaufs betrachtet, der sich aus einer Summation von Sinussignalen sowie aus Sprüngen und Rampen zusammensetzt. Hierdurch soll insbesondere die in Abschnitt 2.4.1 beschriebene und in der Simulation in Abschnitt 3.2.5

[140] Der gesamte Systemzustand $\boldsymbol{x}_k$ und die Stellgröße u_k unter Verwendung der drei hier betrachteten Regler sind in Abbildung D.2 in Anhang D.4 gegeben.

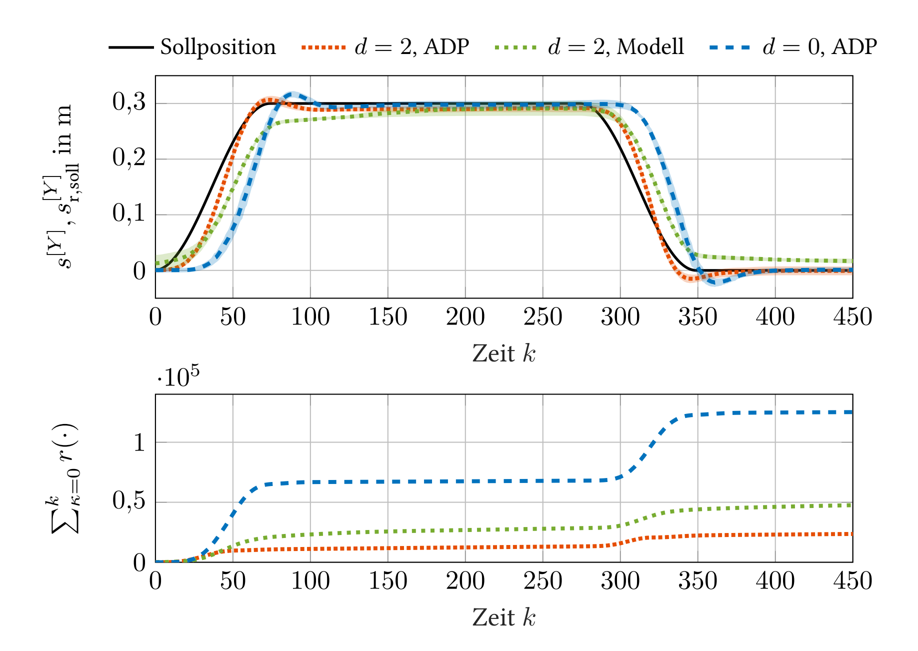


Abbildung 6.18: Vorgabe eines polynomiellen Sollverlaufs mit $d = 2$ für den gelernten, ADP-basierten Regler (rot) sowie den modellbasierten Vergleichsregler (grün) im Vergleich zur Vorgabe einer stationären Sollposition ($d = 0$) für den gelernten, ADP-basierten Regler (blau). *Oben:* Mittlere Ballposition und Standardabweichung über 18, 12, bzw. 11 Wiederholungen. *Unten:* Mittlere akkumulierte Einschrittkosten.

bereits gezeigte Flexibilität des neuartigen ADP-basierten Solltrajektorienfolgereglers verdeutlicht werden, die auch in der realen Anwendung Gültigkeit besitzt. Der gelernte Regler kann unterschiedlichen Solltrajektorienvorgaben erfolgreich folgen, ohne dass ein erneuter Trainingsvorgang benötigt wird. In Abbildung 6.19 ist neben der Solltrajektorie der resultierende Verlauf für den gelernten ADP-Solltrajektorienfolgeregler ($d = 2$) sowie für den ADP-Sollzustandsregler ($d = 0$) gezeigt[141]. Auch hier offenbaren sich die Vorteile des ADP-Solltrajektorienfolgereglers, der den Sollverlauf explizit einbezieht, im Gegensatz zur konstanten Sollzustandsvorgabe. Ersterer kann, aufgrund der lokalen Solltrajektorienapproximation, dem Sollverlauf besser folgen, wodurch insbesondere deutlich geringere akkumulierte Kosten entstehen.

[141] Der gesamte Systemzustand $\boldsymbol{x}_k$ und die Stellgröße u_k sind in Abbildung D.3 in Anhang D.4 gezeigt.

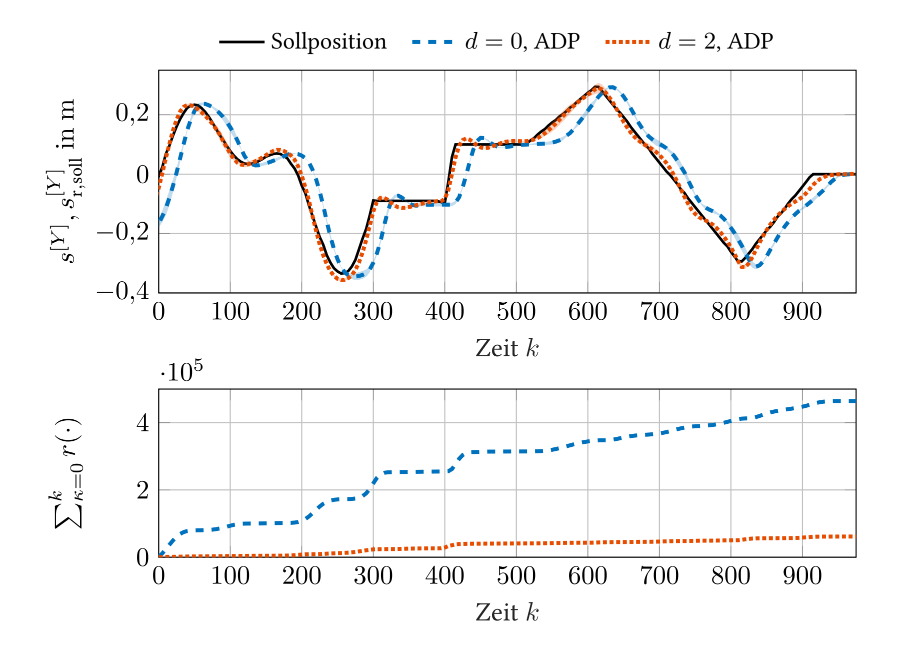


Abbildung 6.19: Vergleich der gelernten Solltrajektorienfolgeregler bei Vorgabe eines polynomiellen Sollverlaufs ($d = 2$, rot) und einer stationären Sollposition ($d = 0$, blau) für eine Validierungstrajektorie. *Oben:* Mittlere Ballposition und Standardabweichung über 4 Wiederholungen. *Unten:* Mittlere akkumulierte Einschrittkosten.

6.2.3.4 Offsetkorrektur und zweidimensionale Regelung mit Approximation des Sollverlaufs

Der Regler, der die X-Dimension des Ball-auf-Platte-Systems regelt, wird mit derselben Methodik und denselben Parametern wie der Regler für die Y-Dimension trainiert. Daraus ergibt sich die durch

$$\boldsymbol{K}_{\text{ADP}}^{[X]} = [\underbrace{65{,}3 \quad 37{,}0 \quad 135{,}1 \quad 18{,}6}_{\boldsymbol{K}_{\text{x}}} \quad \underbrace{-28{,}8 \quad -38{,}2 \quad -61{,}2}_{\boldsymbol{K}_{\text{z}}} \quad \underbrace{2{,}2}_{\boldsymbol{K}_{\text{off}}}] \tag{6.40}$$

gegebene gelernte Reglermatrix. Aufgrund einer nahezu symmetrischen Masseverteilung der Platte in Y-Richtung findet in $\boldsymbol{K}_{\text{ADP}}^{[Y]}$ (6.37) nahezu keine Offsetkorrektur statt. Demgegenüber findet in X-Richtung aufgrund von $K_{\text{off}}^{[X]} = 2{,}2$ in (6.40) eine statische Offsetkorrektur in Form eines konstanten Ausgleichsstroms von $-2{,}2$ A statt, um ein konstruktions- oder fertigungsbedingtes Ungleichgewicht auszugleichen. Im Fall des modellbasierten Vergleichsreglers muss solch ein Ausgleichsstrom in der Regel heuristisch ermittelt werden (vgl. [KIB$^+$19]), da potenzielle Ungleichgewichte und Fertigungstoleranzen meist nicht präzise im Systemmodell berücksichtigt sind. Wie wichtig jedoch die Verwendung eines solchen Ausgleichsstroms ist, zeigt Abbildung 6.20[142]. Bei Verwendung einer Q-Function-Approximation, die keine

[142] Der gesamte Systemzustand $\boldsymbol{x}_k$ sowie die Stellgröße u_k sind in Abbildung D.4 in Anhang D.4 gezeigt.

Offsetkorrektur ermöglicht, kommt es zu einem asymmetrischen Verhalten der Ballposition und somit zu signifikanten Abweichungen von der Sollposition. Im Vergleich dazu führt der gelernte Offsetterm zu einem symmetrischen Verhalten.

Nach Satz 3.1 ergibt sich die Reglermatrix

$$\boldsymbol{K}_{\text{Modell}}^{[X]} = [\underbrace{75{,}9 \quad 56{,}8 \quad 217{,}0 \quad 34{,}8}_{\boldsymbol{K}_{\text{x}}} \quad \underbrace{-45{,}3 \quad -56{,}8 \quad -75{,}4}_{\boldsymbol{K}_{\text{z}}}] \tag{6.41}$$

des modellbasierten Vergleichsreglers, für den ebenfalls ein (in diesem Fall heuristisch ermittelter) Ausgleichsstrom von $-2{,}2$ A verwendet wird. Die gleichzeitige Vorgabe eines Solltrajektorienverlaufs für die Position des Balls in beiden Plattendimensionen ist schließlich sowohl für den ADP-basierten Regler als auch für den Vergleichsregler in Abbildung 6.21 gezeigt. Die zugehörigen Zustands- und Stellgrößenverläufe sind in Abbildung D.5 und Abbildung D.6 (Anhang D.4) gegeben.

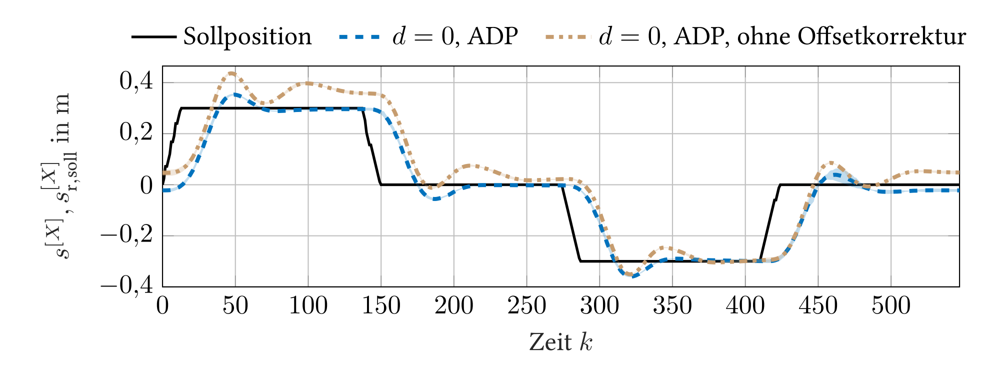


Abbildung 6.20: Vergleich eines gelernten Reglers mit (blau) und ohne (braun) lernbare Offsetkorrektur bei Vorgabe einer stationären Sollposition ($d = 0$).

6.2.4 Diskussion

In diesem Unterkapitel wurde ein modellfreier, ADP-basierter Solltrajektorienfolgeregler für ein reales Ball-auf-Platte-System vorgestellt. Mit weniger als einer Minute aufgezeichneter Messdaten konnte ein Regler trainiert werden, der gegebenen Solltrajektorienverläufen folgen kann und dabei hinsichtlich des zugrunde liegenden Gütefunktionals einem modellbasierten Vergleichsregler überlegen ist. Letzteres resultiert aus einer etwas schnelleren Sprungantwort sowie einem geringeren statischen Fehler: Während der modellbasierte Vergleichsregler Abweichungen vom Systemmodell nicht berücksichtigen kann, verwendet der ADP-Ansatz reale Messdaten und kann so, im Rahmen der Approximationsfähigkeit der verwendeten Basis-

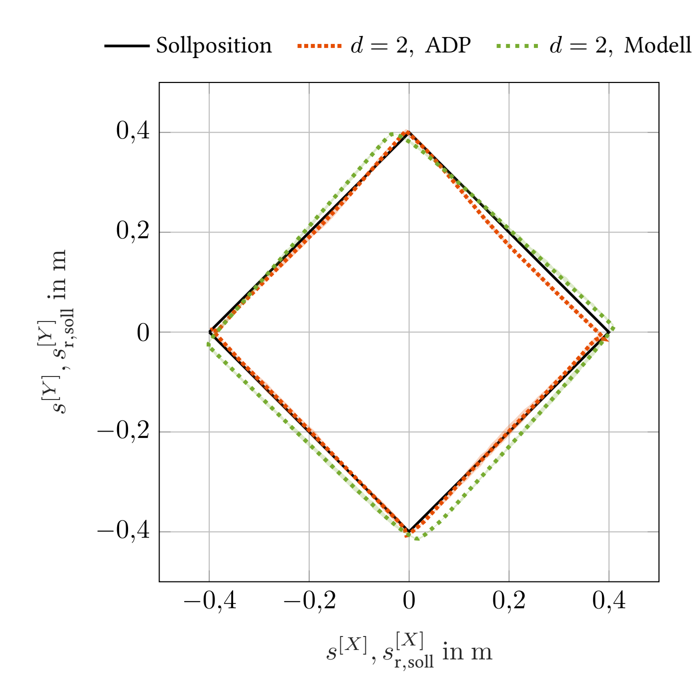


Abbildung 6.21: Gleichzeitige Vorgabe eines Solltrajektorienverlaufs in beiden Plattendimensionen und resultierende Ballposition für den gelernten Regler und den modellbasierten Vergleichsregler mit jeweils $d = 2$. Gezeigt sind die mittlere Ballposition sowie die Standardabweichung über jeweils 3 Wiederholungen.

funktionen, das Systemverhalten bestmöglich im Sinne der zu minimierenden Gütefunktion berücksichtigen[143].

Die Verwendung einer ADP-kompatiblen lokalen Approximation des Sollzustandsverlaufs anstelle einer stationären Sollzustandsvorgabe ermöglicht ein vorausschauendes Verhalten des Reglers, einen verringerten zeitlichen Versatz des Zustands zum Sollzustand und letztlich reduzierte Kosten.

Die experimentellen Ergebnisse zeigen letztlich auch anhand des zweiten realen Anwendungsbeispiels eine grundsätzliche Anwendbarkeit der in dieser Arbeit vorgestellten ADP-kompatiblen Solltrajektorienfolgeregelungsmethoden auf reale, regelungstechnische Problemstellungen. Aufgrund der messdatenbasierten Adaption der Gewichte der Q-Function-Approximation und somit des Regelgesetzes ist weder eine aufwendige exakte Modellbildung noch eine manuelle Feinabstimmung notwendig.

[143] Die Basisfunktionen des ADP-Ansatzes sind aus Gründen der Fairness gegenüber dem Vergleichsregler so gewählt, dass die resultierenden Regelgesetze dieselbe Form aufweisen. Dies ist in (6.37) und (6.38) zu erkennen, wobei anzumerken sei, dass auch im Fall des modellbasierten Reglers ein Ausgleichsstrom, der jedoch heuristisch ermittelt wurde, verwendet wird.

Schließlich soll noch die Wahl der Basisfunktionen zur Approximation der Q-Function diskutiert werden. Der vorgestellte Ansatz kann ohne Weiteres um zusätzliche Basisfunktionen erweitert werden, um beispielsweise auch Nichtlinearitäten im Systemverhalten abbilden zu können. Dies würde jedoch zu einer größeren Anzahl h an Gewichten und somit einer höheren Komplexität der Policy Iteration (vgl. Abschnitt 6.2.2.3) führen. Neben einem höheren Rechenaufwand wären zudem vor allem mehr Trainingstupel M und somit mehr reale Messdaten benötigt. Erfahrungsgemäß erschwert dies auch eine geeignete Anregung und kann in Kombination mit vorhandenem Messrauschen die Trainingsergebnisse verschlechtern. Somit ist eine angemessene Wahl der Basisfunktionen entscheidend für einen dateneffizienten ADP-Algorithmus. Die Wahl der Basisfunktionen sollte mit Bedacht erfolgen und bietet die Chance, vorhandenes Vorwissen über geeignete Reglerstrukturen oder prinzipielles Systemverhalten zu integrieren. Am Beispiel des betrachteten Ball-auf-Platte-Systems wurde auf diese Weise Vorwissen über geeignete Basisfunktionen eingebracht und ein erlernbarer statischer Ausgleichsstrom verwendet. Die Nutzung dieses Vorwissens auf der einen Seite und die Adaptierbarkeit der Q-Function-Gewichte auf der anderen Seite haben letztlich einen dateneffizienten adaptiven Solltrajektorienfolgeregler ermöglicht.

Zusammenfassend kann konstatiert werden, dass Kapitel 6 die in Abschnitt 2.4.1 formulierte Forschungsfrage 1 aus einer anwendungsorientierten Perspektive betrachtet. Der neuartige, ADP-basierte Trajektorienfolgeregler, der vielfältige und flexible Solltrajektorienvorgaben erlaubt, kann anhand realer Messdaten erfolgreich trainiert werden. Zudem weist die Einbeziehung des approximierten Solltrajektorienverlaufs auch in der realen Anwendung Vorteile gegenüber einer stationären Sollzustandsvorgabe auf.

7 Zusammenfassung

Obwohl selbstlernende, ADP-basierte Regelungsansätze vermehrt in den Fokus der aktuellen Forschung gerückt sind, ist die Integration flexibler Solltrajektorien in diese modellfreien Methoden bislang nur unzureichend gelöst. Zahlreiche regelungstechnische Problemstellungen erfordern jedoch, dass Systemgrößen vorgegebenen und flexiblen Solltrajektorienverläufen optimal im Sinne eines gegebenen Gütefunktionals folgen. Die Analyse bestehender ADP-Ansätze aus der Literatur offenbart, dass diese hauptsächlich stationäre Sollzustandsvorgaben oder globale Vorgaben durch Exosysteme betrachten. Ersteren fehlt die Berücksichtigung des weiteren Solltrajektorienverlaufs, letzteren die Möglichkeit, den Sollverlauf flexibel von außen vorzugeben. Eine zweite bislang in der ADP-Literatur nur unzureichend analysierte Problematik betrifft eine angemessene Systemanregung, um Konvergenz zu gewährleisten. Wenngleich ADP-basierte Ansätze stets eine geeignete Anregung erfordern, finden sich hierzu in der Literatur bislang nur wenige theoretische Analysen.

Die vorliegende Arbeit liefert daher Beiträge zu diesen bislang ungelösten wissenschaftlichen Fragestellungen. Insbesondere wurden für adaptive Optimalregler geeignete Solltrajektoriendarstellungen präsentiert, analysiert und in realen Anwendungsbeispielen umgesetzt. Weiterhin wurden Konvergenzeigenschaften von ADP-Reglern untersucht und Bedingungen an den Systemzustand präsentiert, die eine ausreichende Anregung gewährleisten.

Zunächst wurden Konzepte zur effizienten Einbettung flexibler, zur Laufzeit vorgebbarer Solltrajektorienverläufe in den ADP-Mechanismus entwickelt sowie die Existenz und Stabilität der Lösung der zugehörigen Optimierungsprobleme analysiert. Um Solltrajektorienverläufe in einen modellfreien ADP-Formalismus integrieren zu können, wurde in der vorliegenden Arbeit erstmals der Begriff der ADP-kompatiblen Solltrajektoriendarstellung in zeitdiskreter und zeitkontinuierlicher Form definiert. Als hierfür zentrale Eigenschaft, nicht nur des Systemzustands, sondern insbesondere auch der Beschreibung des Solltrajektorienverlaufs, wurde die Einhaltung der Markov-Bedingung identifiziert. Diese ermöglicht eine adäquate und zeitinvariante Kostenrepräsentation in Form einer erlernbaren Value- oder Q-Function, anhand derer Regelgesetze verbessert werden können. Mithilfe der vorgestellten ADP-kompatiblen Solltrajektorienrepräsentationen wurden modellfreie, adaptive optimale Trajektorienfolgeregler entworfen. Aufgrund der expliziten Abhängigkeit der Value- bzw. Q-Function und somit auch des damit verknüpften Regelgesetzes vom aktuellen Solltrajektorienverlauf ergeben sich zwei wesentliche Vorteile: Erstens behält ein gelerntes Regelgesetz auch bei sich änderndem Solltrajektorienverlauf seine Gültigkeit. Zweitens weisen die gelernten Regler vorausschauendes Verhalten auf und sind somit Vergleichsreglern aus der Literatur mit stationärer Sollzustandsvorgabe oder ohne flexible Solltrajektorienrepräsentation hinsichtlich des Kostenfunktionals überlegen.

Neben diesen Beiträgen zu ADP-kompatiblen Solltrajektorienfolgereglern wurde in der vorliegenden Arbeit die Fragestellung einer geeigneten Systemanregung im ADP-Kontext verfolgt. Hierbei wurden am Beispiel eines zeitkontinuierlichen Nicht-Nullsummen-Differenzialspiels erstmalig allgemeingültige hinreichende Frequenzbedingungen an den Systemzustand hergeleitet. Diese Bedingungen gewährleisten die Erfüllung der PE-Eigenschaft für die häufig verwendete Klasse polynomieller Basisfunktionen zur Funktionsapproximation. Die PE-Eigenschaft ist essenziell für Konvergenzgarantien von ADP-Methoden gegen die optimale Lösung. Die vorgestellten Bedingungen beinhalten zudem Freiheitsgrade, die in der Praxis zur Berücksichtigung anwendungsspezifischer Anforderungen genutzt werden können. Die präsentierten Simulationsergebnisse bestätigen die theoretischen Aussagen und zeigen zudem, dass eine Anregung in Form einer Überlagerung harmonischer Schwingungen zu wesentlich schnellerer Konvergenz der Policy Evaluation führen kann als die Verwendung weißen Gaußschen Rauschens. Hieraus kann schließlich die Handlungsempfehlung abgeleitet werden, Systeme, die mithilfe ADP-basierter Methoden geregelt werden sollen, nicht durch klassische Gaußsche Rauschprozesse, sondern vielmehr durch eine Überlagerung harmonischer Schwingungen oder anderer hauptsächlich niederfrequenter Signale anzuregen. Dadurch können mögliche Tiefpasscharakteristiken des Systems berücksichtigt und eine effiziente Anregung erreicht werden.

Schließlich offenbaren die beiden realen Anwendungsbeispiele die Umsetzbarkeit sowie potenzielle Vorteile der vorgestellten adaptiven optimalen Trajektorienfolgeregler. Die vorliegende Arbeit liefert die erstmalige Anwendung ADP-kompatibler Solltrajektorienfolgeregler, die einen zeitvarianten, von außen vorgebbaren, flexiblen Solltrajektorienverlauf explizit in die Q-Function integrieren und mithilfe realer Messdaten trainiert werden. So konnte ein selbstlernender Geschwindigkeitsregler ohne ein Modell der Longitudinaldynamik in einem Realfahrzeug umgesetzt und mithilfe eines Actor-Critic-Ansatzes online, d. h. während der Fahrt, trainiert werden. Die Verwendung der lokalen Approximation des Sollgeschwindigkeitsprofils gemäß des in dieser Arbeit vorgestellten ADP-kompatiblen Mechanismus lieferte hierbei einen Regler, der einer konstanten Sollgeschwindigkeitsvorgabe bezüglich des Kostenfunktionals überlegen ist. Als zweites Anwendungsbeispiel wurde erstmalig ein ADP-basierter Solltrajektorienfolgeregler an einem realen Ball-auf-Platte-System angewandt. Ohne Kenntnis eines exakten Systemmodells können aus aufgezeichneten Messdaten des Systems die Q-Function des optimalen Trajektorienfolgeregelungsproblems und das zugehörige Regelgesetz erlernt werden. Der vorgestellte selbstlernende Regler übertrifft in experimentellen Untersuchungen den modellbasierten Vergleichsregler und erfordert zudem keine manuelle Feinabstimmung. Auch in diesem Anwendungsbeispiel zeigt die neuartige, ADP-kompatible, lokale Solltrajektorienapproximation Vorteile gegenüber einer stationären Sollzustandsvorgabe. Im Fall der neu vorgestellten Methode folgt der Systemzustand der Solltrajektorie präziser, zudem entstehen signifikant geringere akkumulierte Kosten.

Abschließend lässt sich zusammenfassen, dass die ADP-kompatible Darstellung des lokalen Solltrajektorienverlaufs eine effiziente Approximation der mit der Solltrajektorie verbundenen Gesamtkosten und somit eine erfolgreiche Adaption der Reglergewichte ermöglicht. Die grundsätzliche Anwendbarkeit der entwickelten adaptiven optimalen Solltrajektorienfolge-

regler auf reale regelungstechnische Problemstellungen wurde gezeigt. Bei geeigneter Wahl der Basisfunktionen und angemessener Systemanregung können Solltrajektorienfolgeregler erlernt werden, die ohne aufwendige Modellbildung und ohne vorherige Kenntnis der Systemparameter modellbasierte Vergleichsregler hinsichtlich der Performanz sogar übertreffen können. Dies ist insbesondere darauf zurückzuführen, dass die vorgestellten lernbasierten Ansätze im Gegensatz zu rein modellbasierten Reglerentwurfsverfahren Modellierungsungenauigkeiten und Fertigungstoleranzen im Rahmen der gewählten Funktionsapproximatoren ausgleichen können. Die in dieser Arbeit präsentierten Beiträge schließen somit aktuelle Forschungslücken im Kontext selbstlernender Optimalregler und motivieren eine stärkere interdisziplinäre Verflechtung von Regelungstechnik und Maschinellem Lernen.

A Anhang zu Kapitel 3

A.1 Beweisskizze zu Satz 3.4

Beweis:
Die Grundidee des Beweises basiert auf der Verwendung der dynamischen Programmierung und der Herleitung der Lösung der Q-Function durch Rückwärtsinduktion. Für $i = 1, \ldots, d$ sei die i-te Teilmatrix einer Matrix $\mathbf{\Pi} \in \mathbb{R}^{m \times nd}$ durch

$$\mathbf{\Pi}\left[i\right] := \begin{bmatrix} \mathbf{\Pi}(1,(i-1)n+1) & \cdots & \mathbf{\Pi}(1,ni) \\ \vdots & \ddots & \vdots \\ \mathbf{\Pi}(m,(i-1)n+1) & \cdots & \mathbf{\Pi}(m,ni) \end{bmatrix} \tag{A.1}$$

mit $\mathbf{\Pi}\left[i\right] \in \mathbb{R}^{m \times n}$ definiert. Durch $\iota \in \mathbb{N}_{\geq 0}$ sei zunächst ein Platzhalter gegeben, der später durch η (bzw. $\eta + 1$ im Induktionsschritt) ersetzt wird, wobei $\eta = K - \kappa$ die verbliebenen Zeitschritte auf dem Horizont der Länge K bezeichne. Des Weiteren sei m ein Index mit $m \in \mathbb{N} : m > 1$. Im Folgenden werden einige Kurzschreibweisen definiert. Sei

$$\boldsymbol{X}_\iota^0 := \boldsymbol{X}^0 := \begin{bmatrix} \boldsymbol{I}_n & -\boldsymbol{I}_n \end{bmatrix}, \tag{A.2}$$

$$\boldsymbol{X}_\iota^1 := \sqrt{\gamma}\Big(-\boldsymbol{X}^0[1]\boldsymbol{B}\boldsymbol{G}_\iota \begin{bmatrix} \boldsymbol{F}_\iota & \boldsymbol{K}_\iota^{\iota-1} & \cdots & \boldsymbol{K}_\iota^0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{X}^0[1]\boldsymbol{A} & \boldsymbol{X}^0[2] & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}\Big) \tag{A.3}$$

und

$$\begin{aligned} \boldsymbol{X}_\iota^m := \sqrt{\gamma}\Big(&-\boldsymbol{X}_\iota^{m-1}[1]\boldsymbol{B}\boldsymbol{G}_\iota \begin{bmatrix} \boldsymbol{F}_\iota & \boldsymbol{K}_\iota^{\iota-1} & \cdots & \boldsymbol{K}_\iota^0 \end{bmatrix} \\ &+ \begin{bmatrix} \boldsymbol{X}_\iota^{m-1}[1]\boldsymbol{A} & \mathbf{0} & \boldsymbol{X}_\iota^{m-1}[2] & \cdots & \boldsymbol{X}_\iota^{m-1}[\iota-1] \end{bmatrix}\Big) \end{aligned} \tag{A.4}$$

sowie

$$\boldsymbol{U}_\iota^1 := -\boldsymbol{G}_\iota \begin{bmatrix} \boldsymbol{F}_\iota & \boldsymbol{K}_\iota^{\iota-1} & \cdots & \boldsymbol{K}_\iota^0 \end{bmatrix} \tag{A.5}$$

und

$$\begin{aligned} \boldsymbol{U}_\iota^m := \sqrt{\gamma}\Big(&-\boldsymbol{U}_\iota^{m-1}[1]\boldsymbol{B}\boldsymbol{G}_\iota \begin{bmatrix} \boldsymbol{F}_\iota & \boldsymbol{K}_\iota^{\iota-1} & \cdots & \boldsymbol{K}_\iota^0 \end{bmatrix} \\ &+ \begin{bmatrix} \boldsymbol{U}_\iota^{m-1}[1]\boldsymbol{A} & \mathbf{0} & \boldsymbol{U}_\iota^{m-1}[2] & \cdots & \boldsymbol{U}_\iota^{m-1}[\iota-1] \end{bmatrix}\Big) \end{aligned} \tag{A.6}$$

mit

$$\boldsymbol{M}_\iota := \gamma \boldsymbol{B}^\mathsf{T} \left(\sum_{i=0}^{\iota-2} \left(\boldsymbol{X}_\iota^i[1]\right)^\mathsf{T} \boldsymbol{Q} \boldsymbol{X}_\iota^i[1] + \sum_{i=1}^{\iota-2} \left(\boldsymbol{U}_\iota^i[1]\right)^\mathsf{T} \boldsymbol{R} \boldsymbol{U}_\iota^i[1] \right), \tag{A.7}$$

$$\boldsymbol{F}_\iota := \boldsymbol{M}_\iota \boldsymbol{A}, \tag{A.8}$$

$$\boldsymbol{G}_\iota^{-1} := \boldsymbol{M}_\iota \boldsymbol{B} + \boldsymbol{R}, \tag{A.9}$$

$$\boldsymbol{K}_\iota^j := \begin{cases} \gamma \boldsymbol{B}^\mathsf{T} \left(\boldsymbol{X}^0[1]\right)^\mathsf{T} \boldsymbol{Q} \boldsymbol{X}^0[2], & \text{für } j = \iota - 1, \\ \gamma \boldsymbol{B}^\mathsf{T} \left(\sum_{i=1}^{\iota-2} \left(\boldsymbol{X}_\iota^i[1]\right)^\mathsf{T} \boldsymbol{Q} \boldsymbol{X}_\iota^i[\iota - j] + \sum_{i=1}^{\iota-2} \left(\boldsymbol{U}_\iota^i[1]\right)^\mathsf{T} \boldsymbol{R} \boldsymbol{U}_\iota^i[\iota - j] \right), & \text{für } j < \iota - 1, \end{cases} \tag{A.10}$$

wobei $j \in \mathbb{N}_{\geq 0}$. Sei weiterhin

$$\boldsymbol{\rho}_0^\kappa := \begin{bmatrix} \boldsymbol{x}_{k_\kappa}^\mathsf{T} & \boldsymbol{x}_{\mathrm{r},k_\kappa}^\mathsf{T} \end{bmatrix} \left(\boldsymbol{X}^0\right)^\mathsf{T}, \tag{A.11}$$

$$\boldsymbol{\rho}_1^\kappa := \begin{bmatrix} \boldsymbol{x}_{k_\kappa}^\mathsf{T} & \boldsymbol{u}_{k_\kappa}^\mathsf{T} & \boldsymbol{x}_{\mathrm{r},k_\kappa+1}^\mathsf{T} \end{bmatrix} \begin{bmatrix} \left(\boldsymbol{X}^0[1]\boldsymbol{A}\right)^\mathsf{T} \\ \left(\boldsymbol{X}^0[1]\boldsymbol{B}\right)^\mathsf{T} \\ \left(\boldsymbol{X}^0[2]\right)^\mathsf{T} \end{bmatrix}, \tag{A.12}$$

$$\boldsymbol{\mu}_i^\kappa := \begin{bmatrix} \boldsymbol{x}_{k_\kappa}^\mathsf{T} & \boldsymbol{u}_{k_\kappa}^\mathsf{T} & \boldsymbol{x}_{\mathrm{r},k_\kappa+2}^\mathsf{T} & \cdots & \boldsymbol{x}_{\mathrm{r},k+K}^\mathsf{T} \end{bmatrix} \begin{bmatrix} \left(\boldsymbol{U}_\eta^{\eta-i}[1]\boldsymbol{A}\right)^\mathsf{T} \\ \left(\boldsymbol{U}_\eta^{\eta-i}[1]\boldsymbol{B}\right)^\mathsf{T} \\ \left(\boldsymbol{U}_\eta^{\eta-i}[2]\right)^\mathsf{T} \\ \vdots \\ \left(\boldsymbol{U}_\eta^{\eta-i}[\eta]\right)^\mathsf{T} \end{bmatrix}, \tag{A.13}$$

$$\boldsymbol{\chi}_i^\kappa := \begin{bmatrix} \boldsymbol{x}_{k_\kappa}^\mathsf{T} & \boldsymbol{u}_{k_\kappa}^\mathsf{T} & \boldsymbol{x}_{\mathrm{r},k_\kappa+2}^\mathsf{T} & \cdots & \boldsymbol{x}_{\mathrm{r},k+K}^\mathsf{T} \end{bmatrix} \begin{bmatrix} \left(\boldsymbol{X}_\eta^{\eta-i}[1]\boldsymbol{A}\right)^\mathsf{T} \\ \left(\boldsymbol{X}_\eta^{\eta-i}[1]\boldsymbol{B}\right)^\mathsf{T} \\ \left(\boldsymbol{X}_\eta^{\eta-i}[2]\right)^\mathsf{T} \\ \vdots \\ \left(\boldsymbol{X}_\eta^{\eta-i}[\eta]\right)^\mathsf{T} \end{bmatrix} \tag{A.14}$$

mit $k_\kappa = k + K - \eta = k + \kappa$ und $i \in \mathbb{N}$.

Nachfolgend wird mittels Rückwärtsinduktion gezeigt, dass die zu dem System (3.80) und dem Gütefunktional (3.82) gehörende Q-Function ${}^K\!Q_\kappa$ nach Definition 3.6 durch

$$\begin{aligned} {}^K\!Q_\kappa = \frac{1}{2} \Bigg(& \boldsymbol{\rho}_0^\kappa \boldsymbol{Q} \left(\boldsymbol{\rho}_0^\kappa\right)^\mathsf{T} + \boldsymbol{u}_{k+\kappa}^\mathsf{T} \boldsymbol{R} \boldsymbol{u}_{k+\kappa} + \gamma \boldsymbol{\rho}_1^\kappa \boldsymbol{Q} \left(\boldsymbol{\rho}_1^\kappa\right)^\mathsf{T} \\ & + \gamma \sum_{i=1}^{K-\kappa-2} \left(\boldsymbol{\chi}_i^\kappa \boldsymbol{Q} \left(\boldsymbol{\chi}_i^\kappa\right)^\mathsf{T} + \boldsymbol{\mu}_i^\kappa \boldsymbol{R} \left(\boldsymbol{\mu}_i^\kappa\right)^\mathsf{T} \right) \Bigg) \end{aligned} \tag{A.15}$$

gegeben ist. Ausgehend von ${}^K\!Q_K$ (vgl. (3.92)) folgt aus Definition 3.6 und

$$\left.\frac{\partial\, {}^K\!Q_K}{\partial \boldsymbol{u}_{k+K}}\right|_{\boldsymbol{u}_{k+K}^*} = \boldsymbol{0}, \qquad \left.\frac{\partial^2\, {}^K\!Q_K}{\partial \boldsymbol{u}_{k+K}^2}\right|_{\boldsymbol{u}_{k+K}^*} = \boldsymbol{R} \succ \boldsymbol{0} \tag{A.16}$$

direkt $\boldsymbol{u}^*_{k+K} = \boldsymbol{0}$. Durch rückwärtige Iteration über die Zeit und Verwendung von (3.91) sowie der Systemdynamik (3.80) kann mit $\eta = K - \kappa$ gezeigt werden, dass (A.15) für $\eta = 0, 1, 2$, d. h. $\kappa = K, K-1, K-2$, gilt. Des Weiteren minimiert

$$\boldsymbol{u}^*_{k+\kappa} = -\boldsymbol{G}_\eta \left(\boldsymbol{F}_\eta \boldsymbol{x}_{k+\kappa} + \sum_{j=0}^{\eta-1} \boldsymbol{K}^j_\eta \boldsymbol{x}_{\mathrm{r},k+K-j} \right) \tag{A.17}$$

den Ausdruck (A.15), weil aufgrund von $\boldsymbol{R} \succ \boldsymbol{0}$ und $\boldsymbol{Q} \succeq \boldsymbol{0}$

$$\left. \frac{\partial\, {}^K Q_\kappa}{\partial \boldsymbol{u}_{k+\kappa}} \right|_{\boldsymbol{u}^*_{k+\kappa}} = \boldsymbol{0}, \quad \text{und} \quad \left. \frac{\partial^2\, {}^K Q_\kappa}{\partial \boldsymbol{u}^2_{k+\kappa}} \right|_{\boldsymbol{u}^*_{k+\kappa}} \succ \boldsymbol{0} \tag{A.18}$$

sichergestellt ist.

Die Induktionsbehauptung ${}^K Q_{\kappa-1}$ (vgl. (A.15) mit $\kappa \to \kappa - 1$) wird dann im Induktionsschritt bewiesen. Hierzu wird ${}^K Q_{\kappa-1}$ durch (3.91) ausgedrückt und $\boldsymbol{u}^*_{k+\kappa}$ aus (A.17) verwendet. Damit folgt

$$\begin{aligned}
{}^K Q_{\kappa-1} = {} & \frac{1}{2} \begin{bmatrix} \boldsymbol{x}_{k+\kappa-1} \\ \boldsymbol{x}_{\mathrm{r},k+\kappa-1} \end{bmatrix}^\intercal \left(\boldsymbol{X}^0\right)^\intercal \boldsymbol{Q} \boldsymbol{X}^0 \begin{bmatrix} \boldsymbol{x}_{k+\kappa-1} \\ \boldsymbol{x}_{\mathrm{r},k+\kappa-1} \end{bmatrix} + \frac{1}{2} \boldsymbol{u}^\intercal_{k+\kappa-1} \boldsymbol{R} \boldsymbol{u}_{k+\kappa-1} \\
& + \frac{1}{2}\gamma \begin{bmatrix} \boldsymbol{x}_{k+\kappa} \\ \boldsymbol{x}_{\mathrm{r},k+\kappa} \end{bmatrix}^\intercal \left(\boldsymbol{X}^0\right)^\intercal \boldsymbol{Q} \boldsymbol{X}^0 \begin{bmatrix} \boldsymbol{x}_{k+\kappa} \\ \boldsymbol{x}_{\mathrm{r},k+\kappa} \end{bmatrix} \\
& + \frac{1}{2}\gamma \bar{\boldsymbol{z}}^\intercal_k \sum_{i=1}^{\eta-1} \left(\left(\boldsymbol{X}^i_{\eta+1}\right)^\intercal \boldsymbol{Q} \boldsymbol{X}^i_{\eta+1} + \left(\boldsymbol{U}^i_{\eta+1}\right)^\intercal \boldsymbol{R} \boldsymbol{U}^i_{\eta+1} \right) \bar{\boldsymbol{z}}_k
\end{aligned} \tag{A.19}$$

mit $\bar{\boldsymbol{z}}^\intercal_k := \begin{bmatrix} \boldsymbol{x}^\intercal_{k+\kappa} & \boldsymbol{x}^\intercal_{\mathrm{r},k+\kappa+1} & \dots & \boldsymbol{x}^\intercal_{\mathrm{r},k+K} \end{bmatrix}$. Einsetzen von $\boldsymbol{x}_{k+\kappa} = \boldsymbol{A}\boldsymbol{x}_{k+\kappa-1} + \boldsymbol{B}\boldsymbol{u}_{k+\kappa-1}$ (vgl. (3.80)) in (A.19) liefert

$$\begin{aligned}
{}^K Q_{\kappa-1} = \frac{1}{2} \Bigg(& \boldsymbol{\rho}^{\kappa-1}_0 \boldsymbol{Q} \left(\boldsymbol{\rho}^{\kappa-1}_0\right)^\intercal + \boldsymbol{u}^\intercal_{k+\kappa-1} \boldsymbol{R} \boldsymbol{u}_{k+\kappa-1} + \gamma \boldsymbol{\rho}^{\kappa-1}_1 \boldsymbol{Q} \left(\boldsymbol{\rho}^{\kappa-1}_1\right)^\intercal \\
& + \gamma \sum_{i=1}^{K-(\kappa-1)-2} \left(\boldsymbol{\chi}^{\kappa-1}_i \boldsymbol{Q} \left(\boldsymbol{\chi}^{\kappa-1}_i\right)^\intercal + \boldsymbol{\mu}^{\kappa-1}_i \boldsymbol{R} \left(\boldsymbol{\mu}^{\kappa-1}_i\right)^\intercal \right) \Bigg)
\end{aligned} \tag{A.20}$$

und somit die Induktionsbehauptung ((A.15) mit $\kappa \to \kappa - 1$). Daher gilt (A.15).

Somit ist die analytische Lösung von ${}^K Q_0$ quadratisch bezüglich $\boldsymbol{\rho}^\kappa_0$, $\boldsymbol{u}_k$, $\boldsymbol{\rho}^\kappa_1$, $\boldsymbol{\chi}^\kappa_i$ und $\boldsymbol{\mu}^\kappa_i$. Da nach (A.11)–(A.14) jede dieser Komponenten linear bezüglich $\boldsymbol{x}_k$, $\boldsymbol{u}_k$ und $\boldsymbol{x}_{\mathrm{r},k+1}, \dots, \boldsymbol{x}_{\mathrm{r},k+n_\mathrm{h}}$ ist, folgt Satz 3.4 direkt für $\kappa = 0$ und $K \geq n_\mathrm{h}$. Die genauen Werte von $\boldsymbol{H}_K$ können aus dem im Beweis gegebenen Schema beispielsweise mithilfe von MATLAB berechnet werden. □

A.2 Beweis zu Lemma 3.8

Beweis:
Nach Lemma 3.7 minimiert die durch $-\boldsymbol{K}\left(\boldsymbol{Z}^{[l]}\right)\begin{bmatrix}\boldsymbol{x}_{k+1}^{\mathsf{T}} & \boldsymbol{x}_{\mathrm{r},k+2}^{\mathsf{T}} & \dots & \boldsymbol{x}_{\mathrm{r},k+n_{\mathrm{h}}}^{\mathsf{T}} & \mathbf{0}^{\mathsf{T}}\end{bmatrix}$ gegebene Stellgröße den Ausdruck $\tilde{\boldsymbol{y}}_{k+1}^{\mathsf{T}}\boldsymbol{Z}^{[l]}\tilde{\boldsymbol{y}}_{k+1}$, $\forall l > 0$. Aufgrund von

$$\boldsymbol{M}\left(-\boldsymbol{K}\left(\boldsymbol{Z}^{[l]}\right)\right)\tilde{\boldsymbol{y}}_k = \tilde{\boldsymbol{y}}_{k+1}\Big|_{\boldsymbol{u}_{k+1}=-\boldsymbol{K}\left(\boldsymbol{Z}^{[l]}\right)\left[\boldsymbol{x}_{k+1}^{\mathsf{T}} \;\; \boldsymbol{x}_{\mathrm{r},k+2}^{\mathsf{T}} \;\; \cdots \;\; \boldsymbol{x}_{\mathrm{r},k+n_{\mathrm{h}}}^{\mathsf{T}} \;\; \mathbf{0}^{\mathsf{T}}\right]} \tag{A.21}$$

gilt daher

$$\begin{aligned}&\tilde{\boldsymbol{y}}_k^{\mathsf{T}}\boldsymbol{M}\left(\boldsymbol{W}^{[l]}\right)^{\mathsf{T}}\boldsymbol{Z}^{[l]}\boldsymbol{M}\left(\boldsymbol{W}^{[l]}\right)\tilde{\boldsymbol{y}}_k\\ \geq&\tilde{\boldsymbol{y}}_k^{\mathsf{T}}\boldsymbol{M}\left(-\boldsymbol{K}\left(\boldsymbol{Z}^{[l]}\right)\right)^{\mathsf{T}}\boldsymbol{Z}^{[l]}\boldsymbol{M}\left(-\boldsymbol{K}\left(\boldsymbol{Z}^{[l]}\right)\right)\tilde{\boldsymbol{y}}_k.\end{aligned} \tag{A.22}$$

Daraus folgt

$$\boldsymbol{M}\left(\boldsymbol{W}^{[l]}\right)^{\mathsf{T}}\boldsymbol{Z}^{[l]}\boldsymbol{M}\left(\boldsymbol{W}^{[l]}\right) - \boldsymbol{M}\left(-\boldsymbol{K}\left(\boldsymbol{Z}^{[l]}\right)\right)^{\mathsf{T}}\boldsymbol{Z}^{[l]}\boldsymbol{M}\left(-\boldsymbol{K}\left(\boldsymbol{Z}^{[l]}\right)\right) \succeq \mathbf{0} \tag{A.23}$$

und somit für die nach (3.129) definierte Folge

$$\boldsymbol{Z}^{[l+1]} = F\left(\boldsymbol{Z}^{[l]}, \boldsymbol{W}^{[l]}\right) \succeq F\left(\boldsymbol{Z}^{[l]}, -\boldsymbol{K}\left(\boldsymbol{Z}^{[l]}\right)\right) =: \hat{\boldsymbol{Z}}^{[l+1]}. \tag{A.24}$$

Daher gilt auch

$$F\left(\boldsymbol{H}^{[l]}, -\boldsymbol{K}\left(\boldsymbol{Z}^{[l]}\right)\right) \succeq F\left(\boldsymbol{H}^{[l]}, -\boldsymbol{K}\left(\boldsymbol{H}^{[l]}\right)\right) = \boldsymbol{H}^{[l+1]}. \tag{A.25}$$

Mit dem Induktionsanfang $\mathbf{0} \preceq \boldsymbol{H}^{[0]} \preceq \boldsymbol{Z}^{[0]}$ und der Induktionsbehauptung $\boldsymbol{H}^{[l]} \preceq \boldsymbol{Z}^{[l]}$ ergibt sich

$$\begin{aligned}F\left(\boldsymbol{H}^{[l]}, -\boldsymbol{K}\left(\boldsymbol{Z}^{[l]}\right)\right) &= \boldsymbol{G} + \gamma\boldsymbol{M}\left(-\boldsymbol{K}\left(\boldsymbol{Z}^{[l]}\right)\right)^{\mathsf{T}}\boldsymbol{H}^{[l]}\boldsymbol{M}\left(-\boldsymbol{K}\left(\boldsymbol{Z}^{[l]}\right)\right)\\ &\preceq \boldsymbol{G} + \gamma\boldsymbol{M}\left(-\boldsymbol{K}\left(\boldsymbol{Z}^{[l]}\right)\right)^{\mathsf{T}}\boldsymbol{Z}^{[l]}\boldsymbol{M}\left(-\boldsymbol{K}\left(\boldsymbol{Z}^{[l]}\right)\right)\\ &= F\left(\boldsymbol{Z}^{[l]}, -\boldsymbol{K}\left(\boldsymbol{Z}^{[l]}\right)\right) = \hat{\boldsymbol{Z}}^{[l+1]}.\end{aligned} \tag{A.26}$$

Zusammen mit (A.25) folgt daraus $\boldsymbol{H}^{[l+1]} \preceq \hat{\boldsymbol{Z}}^{[l+1]}$ und unter Einbeziehung von (A.24) ergibt sich schließlich

$$\mathbf{0} \preceq \boldsymbol{H}^{[l+1]} \preceq \boldsymbol{Z}^{[l+1]}. \tag{A.27}$$

□

A.3 Beweis zu Lemma 3.9

Beweis:
Der Beweis stellt eine Erweiterung des Beweises von [Lan97, Lemma B.1.2] auf den Solltrajektorienfolgeregelungsfall dar. Sei

$$\boldsymbol{Z}^{[l+1]} = F\left(\boldsymbol{Z}^{[l]}, -\tilde{\boldsymbol{K}}\right) \tag{A.28}$$

mit $\boldsymbol{Z}^{[0]} = \boldsymbol{H}^{[0]}$ gegeben, wobei $\tilde{\boldsymbol{K}}$ so gewählt ist, dass alle Eigenwerte von $\left(\boldsymbol{A} - \boldsymbol{B}\tilde{\boldsymbol{K}}_x\right)$ innerhalb des Einheitskreises liegen. Die Existenz von $\tilde{\boldsymbol{K}}$ ist aufgrund der Steuerbarkeit von $(\boldsymbol{A}, \boldsymbol{B})$ gewährleistet. Mit $\boldsymbol{W}^{[l]} = -\tilde{\boldsymbol{K}}$ folgt aus Lemma 3.8, dass

$$\boldsymbol{0} \preceq \boldsymbol{H}^{[l]} \preceq \boldsymbol{Z}^{[l]} \tag{A.29}$$

gilt. Aus

$$\begin{aligned}\boldsymbol{Z}^{[l+1]} - \boldsymbol{Z}^{[l]} &= F\left(\boldsymbol{Z}^{[l]}, -\tilde{\boldsymbol{K}}\right) - F\left(\boldsymbol{Z}^{[l-1]}, -\tilde{\boldsymbol{K}}\right) \\ &= \gamma \boldsymbol{M}\left(-\tilde{\boldsymbol{K}}\right)^{\mathsf{T}} \left(\boldsymbol{Z}^{[l]} - \boldsymbol{Z}^{[l-1]}\right) \boldsymbol{M}\left(-\tilde{\boldsymbol{K}}\right)\end{aligned} \tag{A.30}$$

folgt[144]

$$\operatorname{vec}\left(\boldsymbol{Z}^{[l+1]} - \boldsymbol{Z}^{[l]}\right) = \underbrace{\gamma \boldsymbol{M}\left(-\tilde{\boldsymbol{K}}\right)^{\mathsf{T}} \otimes \boldsymbol{M}\left(-\tilde{\boldsymbol{K}}\right)}_{=:\boldsymbol{E}} \operatorname{vec}\left(\boldsymbol{Z}^{[l]} - \boldsymbol{Z}^{[l-1]}\right) \tag{A.31}$$

und somit

$$\operatorname{vec}\left(\boldsymbol{Z}^{[l]} - \boldsymbol{Z}^{[l-1]}\right) = \boldsymbol{E}^{l-1} \operatorname{vec}\left(\boldsymbol{Z}^{[1]} - \boldsymbol{Z}^{[0]}\right). \tag{A.32}$$

Falls nun alle Eigenwerte von $\sqrt{\gamma}\boldsymbol{M}\left(-\tilde{\boldsymbol{K}}\right)$ im Inneren des Einheitskreises liegen, so gilt dies auch für die Eigenwerte von $\boldsymbol{E}$. Aufgrund der in (3.121) gegebenen Struktur folgt, dass mindestens $(n_{\mathrm{h}}+1)n$ Eigenwerte von $\boldsymbol{M}\left(-\tilde{\boldsymbol{K}}\right)$ im Koordinatenursprung liegen. Alle übrigen Eigenwerte, d. h. die Eigenwerte von

$$\bar{\boldsymbol{M}} := \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ -\tilde{\boldsymbol{K}}_{\mathrm{x}}\boldsymbol{A} & -\tilde{\boldsymbol{K}}_{\mathrm{x}}\boldsymbol{B} \end{bmatrix}, \tag{A.33}$$

werden analog zum Vorgehen in [Lan97, Lemma B.1.2] nachfolgend analysiert. Sei $\|\cdot\|_2$ die Spektralnorm einer Matrix bzw. die Euklidische Norm eines Vektors. Dann gilt

$$\begin{aligned}\lim_{l\to\infty} \left\|\bar{\boldsymbol{M}}^l\right\|_2 &= \lim_{l\to\infty} \left\| \begin{bmatrix} \boldsymbol{I}_n \\ -\tilde{\boldsymbol{K}}_{\mathrm{x}} \end{bmatrix} \left(\boldsymbol{A} - \boldsymbol{B}\tilde{\boldsymbol{K}}_{\mathrm{x}}\right)^{l-1} \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \end{bmatrix} \right\|_2 \\ &\le \lim_{l\to\infty} \left\| \begin{bmatrix} \boldsymbol{I}_n \\ -\tilde{\boldsymbol{K}}_{\mathrm{x}} \end{bmatrix} \right\|_2 \left\| \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \end{bmatrix} \right\|_2 \left\| \left(\boldsymbol{A} - \boldsymbol{B}\tilde{\boldsymbol{K}}_{\mathrm{x}}\right)^{l-1} \right\|_2 = 0.\end{aligned}$$

[144] Der Operator vec(·) nimmt eine vertikale Konkatenation der Spalten einer Matrix vor und ⊗ bezeichnet das Kronecker-Produkt.

Aus $\lim_{l\to\infty} \bar{\boldsymbol{M}}^l = \boldsymbol{0}$ folgt, dass alle Eigenwerte von $\bar{\boldsymbol{M}}$ im Inneren des Einheitskreises liegen. Folglich sind alle Eigenwerte von $\boldsymbol{E}$ ebenfalls innerhalb des Einheitskreises und es gilt $e := \|\boldsymbol{E}\|_2 < 1$. Aus

$$\begin{aligned}\mathrm{vec}\left(\boldsymbol{Z}^{[j]}\right) &= \mathrm{vec}\left(\boldsymbol{Z}^{[0]}\right) + \sum_{l=1}^{j} \mathrm{vec}\left(\boldsymbol{Z}^{[l]} - \boldsymbol{Z}^{[l-1]}\right) \\ &\overset{(A.32)}{=} \mathrm{vec}\left(\boldsymbol{Z}^{[0]}\right) + \sum_{l=1}^{j} \boldsymbol{E}^{l-1} \mathrm{vec}\left(\boldsymbol{Z}^{[1]} - \boldsymbol{Z}^{[0]}\right)\end{aligned} \tag{A.34}$$

folgt

$$\left\|\mathrm{vec}\left(\boldsymbol{Z}^{[j]}\right)\right\|_2 \leq \left\|\mathrm{vec}\left(\boldsymbol{Z}^{[0]}\right)\right\|_2 + \sum_{l=0}^{\infty} \|\boldsymbol{E}\|_2^l \left\|\mathrm{vec}\left(\boldsymbol{Z}^{[1]} - \boldsymbol{Z}^{[0]}\right)\right\|_2 =: e_0, \tag{A.35}$$

wobei die obere Schranke e_0 unabhängig von j ist. Da $\left\|\mathrm{vec}\left(\boldsymbol{Z}^{[j]}\right)\right\|_2$ durch e_0 nach oben beschränkt ist, existiert e_1, sodass $\left\|\boldsymbol{Z}^{[j]}\right\|_2 \leq e_1, \forall j$. Mit $\boldsymbol{Y} := e_1 \boldsymbol{I}_{\mathrm{Dim}(\boldsymbol{H})}$ ergibt sich schließlich

$$\boldsymbol{0} \preceq \boldsymbol{H}^{[l]} \preceq \boldsymbol{Z}^{[l]} \preceq \left\|\boldsymbol{Z}^{[l]}\right\|_2 \boldsymbol{I}_{\mathrm{Dim}(\boldsymbol{H})} \preceq e_1 \boldsymbol{I}_{\mathrm{Dim}(\boldsymbol{H})} = \boldsymbol{Y} \tag{A.36}$$

und somit die Aussage von Lemma 3.9. □

A.4 Ergänzungen zum linearen Einspurmodell

Die physikalischen Parameter sowie das zugehörige zeitkontinuierliche Modell des in (3.143) gegebenen linearen Einspurmodells sind in [Fla16, Anhang B] zu finden. Die Stellgröße u_k entspricht einem auf das Lenkrad aufgebrachten Drehmoment. Der Systemzustand ist durch

$$\boldsymbol{x}_k = \begin{bmatrix}\beta_k & \psi_{\mathrm{r},k} & \psi_k & y_k & \delta_{\mathrm{LR,v},k} & \delta_{\mathrm{LR},k}\end{bmatrix}^\mathsf{T} \tag{A.37}$$

gegeben, wobei β_k der Schwimmwinkel, ψ_k der Gierwinkel, $\psi_{\mathrm{r},k}$ die Gierrate, y_k die laterale Abweichung vom Koordinatenursprung, $\delta_{\mathrm{LR},k}$ der Lenkradwinkel und $\delta_{\mathrm{LR,v},k}$ die Winkelgeschwindigkeit des Lenkrads ist. Die geometrischen Zusammenhänge sind Abbildung A.1 zu entnehmen[145], hierbei stellt v die als konstant angenommene Geschwindigkeit des Fahrzeugs dar. Im vorliegenden Beispiel sei $v = 20\,\mathrm{m\,s^{-1}}$. Zudem stellt $\delta_\mathrm{s} = 0{,}0625\delta_\mathrm{LR}$ den Lenkwinkel und dessen Zusammenhang zum Lenkradwinkel δ_LR dar.

[145] Für eine ausführliche Diskussion des Modells sei auf [Fla16, Anhang B] verwiesen.

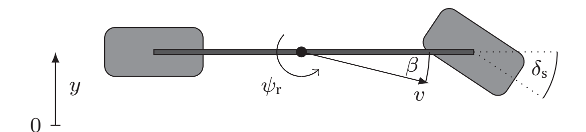


Abbildung A.1: Geometrische Zusammenhänge des linearen Einspurmodells.

A.5 ADP-Solltrajektorienregler für das lineare Einspurmodell mit $\gamma = 1$

Für das Beispiel des linearen Einspurmodells (3.143) mit den in (3.144) gegebenen Gütemaßparametern und $\gamma = 1$ sind die resultierenden Trajektorienverläufe in Abbildung A.2 und die Gewichtsfehler in Abbildung A.3 gegeben.

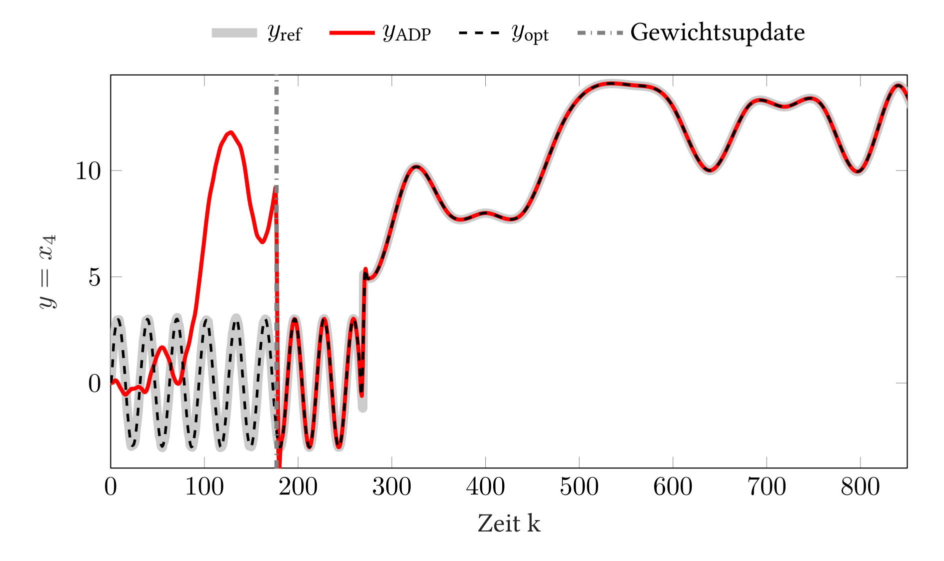


Abbildung A.2: Ergebnis der ADP-Solltrajektorienregelung für System 2 (lineares Einspurmodell sechster Ordnung) für $\gamma = 1$.

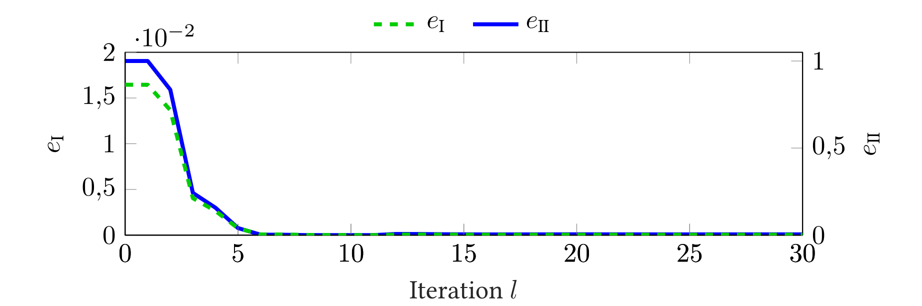


Abbildung A.3: Gewichtsfehlerverlauf während der ersten 30 Iterationen des Lernvorgangs für System 2 (lineares Einspurmodell sechster Ordnung) für $\gamma = 1$. Hierbei stellt e_{I} (3.148) den Mittelwert und e_{II} (3.149) das Maximum des elementweisen absoluten Fehlers von $\hat{\boldsymbol{w}}$, jeweils durch $\max_j \left|\{\boldsymbol{w}^*\}_j\right|$ normiert, dar.

B Anhang zu Kapitel 4

B.1 Beweis zu Lemma 4.1

Beweis:
Betrachtet werde $x_{\mathrm{r},i}(t)$ nach (4.7). Sei $J_{i,\mathrm{r}}$ die Anzahl verschiedener reeller Eigenwerte sowie $J_{i,\mathrm{c}}$ die Anzahl verschiedener konjugiert-komplexer Eigenwertpaare, d. h. $J_i = J_{i,\mathrm{r}} + 2J_{i,\mathrm{c}}$. Dann folgt mit Annahme 4.1

$$x_{\mathrm{r},i}(t) = \sum_{j=1}^{J_{i,\mathrm{r}}} \left(\sum_{k=0}^{\nu_{ij}-1} c_{ijk} t^k \right) e^{\lambda_{ij} t} \tag{B.1}$$

$$+ \sum_{j=J_{i,\mathrm{r}}+1}^{J_{i,\mathrm{r}}+J_{i,\mathrm{c}}} \left(\sum_{k=0}^{\nu_{ij}-1} c_{ijk} t^k \right) e^{\lambda_{ij} t} + \sum_{j=J_{i,\mathrm{r}}+J_{i,\mathrm{c}}+1}^{J_i} \left(\sum_{k=0}^{\nu_{ij}-1} c^\star_{ijk} t^k \right) e^{\lambda^\star_{ij} t} \tag{B.2}$$

$$= \mathrm{Re}\left\{ \sum_{j=1}^{J_{i,\mathrm{r}}} \left(\sum_{k=0}^{\nu_{ij}-1} c_{ijk} t^k \right) e^{\lambda_{ij} t} \right\} + 2\,\mathrm{Re}\left\{ \sum_{j=J_{i,\mathrm{r}}+1}^{J_{i,\mathrm{r}}+J_{i,\mathrm{c}}} \left(\sum_{k=0}^{\nu_{ij}-1} c_{ijk} t^k \right) e^{\lambda_{ij} t} \right\} \tag{B.3}$$

und somit Reellwertigkeit von $x_{\mathrm{r},i}(t)$ und $\boldsymbol{x}_\mathrm{r}(t)$. □

B.2 Beweis zu Lemma 4.2

Beweis:
Der Beweis erfolgt mittels vollständiger Induktion. Für den Induktionsanfang ergibt sich

$$x^{(0)}_{\mathrm{r},ij}(t) = \mathrm{e}^{\lambda_{ij} t} \binom{0}{0} \lambda^0_{ij} \sum_{k=0}^{\nu_{ij}-1} \frac{k!}{k!} c_{ijk} t^j = \mathrm{e}^{\lambda_{ij} t} \sum_{k=0}^{\nu_{ij}-1} c_{ijk} t^j = x_{\mathrm{r},ij}(t). \tag{B.4}$$

Im Induktionsschritt folgt aus der zeitlichen Ableitung des Ausdrucks $x^{(l)}_{\mathrm{r},ij}(t)$ in (4.23)

$$x^{(l+1)}_{\mathrm{r},ij}(t) = \sum_{m=0}^{l} \mathrm{e}^{\lambda_{ij} t} \binom{l}{m} \lambda^{l+1-m}_{ij} \sum_{k=m}^{\nu_{ij}-1} \frac{k!}{(k-m)!} c_{ijk} t^{k-m}$$

$$+ \sum_{m=0}^{l} \mathrm{e}^{\lambda_{ij} t} \binom{l}{m} \lambda_{ij}^{l-m} \sum_{k=m+1}^{\nu_{ij}-1} \frac{k!}{(k-m)!} (k-m) c_{ijk} t^{k-(m+1)} \tag{B.5}$$

$$= \mathrm{e}^{\lambda_{ij} t} \binom{l}{0} \lambda_{ij}^{l+1-0} \sum_{k=0}^{\nu_{ij}-1} \frac{k!}{(k-0)!} c_{ijk} t^{k-0}$$

$$+ \sum_{m=1}^{l} \mathrm{e}^{\lambda_{ij} t} \binom{l}{m} \lambda_{ij}^{l+1-m} \sum_{k=m}^{\nu_{ij}-1} \frac{k!}{(k-m)!} c_{ijk} t^{k-m}$$

$$+ \sum_{m=0}^{l-1} \mathrm{e}^{\lambda_{ij} t} \binom{l}{m} \lambda_{ij}^{l-m} \sum_{k=m+1}^{\nu_{ij}-1} \frac{k!}{(k-(m+1))!} c_{ijk} t^{k-(m+1)}$$

$$+ \mathrm{e}^{\lambda_{ij} t} \binom{l}{l} \lambda_{ij}^{l-l} \sum_{k=l+1}^{\nu_{ij}-1} \frac{k!}{(k-(l+1))!} c_{ijk} t^{k-(l+1)} \tag{B.6}$$

$$= \mathrm{e}^{\lambda_{ij} t} \binom{l+1}{0} \lambda_{ij}^{l+1-0} \sum_{k=0}^{\nu_{ij}-1} \frac{k!}{(k-0)!} c_{ijk} t^{k-0}$$

$$+ \sum_{m=1}^{l} \mathrm{e}^{\lambda_{ij} t} \binom{l}{m} \lambda_{ij}^{l+1-m} \sum_{k=m}^{\nu_{ij}-1} \frac{k!}{(k-m)!} c_{ijk} t^{k-m}$$

$$+ \sum_{m=1}^{l} \mathrm{e}^{\lambda_{ij} t} \binom{l}{m-1} \lambda_{ij}^{l+1-m} \sum_{k=m}^{\nu_{ij}-1} \frac{k!}{(k-m)!} c_{ijk} t^{k-m} \dots$$

$$\dots + \mathrm{e}^{\lambda_{ij} t} \binom{l+1}{l+1} \lambda_{ij}^{l+1-(l+1)} \sum_{k=l+1}^{\nu_{ij}-1} \frac{k!}{(k-(l+1))!} c_{ijk} t^{k-(l+1)} \tag{B.7}$$

$$= \mathrm{e}^{\lambda_{ij} t} \binom{l+1}{0} \lambda_{ij}^{l+1-0} \sum_{k=0}^{\nu_{ij}-1} \frac{k!}{(k-0)!} c_{ijk} t^{k-0}$$

$$+ \sum_{m=1}^{l} \mathrm{e}^{\lambda_{ij} t} \left[\binom{l}{m-1} + \binom{l}{m} \right] \lambda_{ij}^{l+1-m} \sum_{k=m}^{\nu_{ij}-1} \frac{k!}{(k-m)!} c_{ijk} t^{k-m}$$

$$+ \mathrm{e}^{\lambda_{ij} t} \binom{l+1}{l+1} \lambda_{ij}^{l+1-(l+1)} \sum_{k=l+1}^{\nu_{ij}-1} \frac{k!}{(k-(l+1))!} c_{ijk} t^{k-(l+1)} \tag{B.8}$$

$$= \sum_{m=0}^{l+1} \mathrm{e}^{\lambda_{ij} t} \binom{l+1}{m} \lambda_{ij}^{l+1-m} \sum_{k=m}^{\nu_{ij}-1} \frac{k!}{(k-m)!} c_{ijk} t^{k-m}, \tag{B.9}$$

wobei im letzten Schritt die Beziehung $\binom{l}{m-1} + \binom{l}{m} = \binom{l+1}{m}$ verwendet wird. Der Ausdruck (B.9) entspricht der Induktionsbehauptung, d. h. (4.23) mit $l+1$. □

B.3 Zustandstransformation für reellwertige $\boldsymbol{\zeta}(t)$ und $\boldsymbol{D}$

Für jedes konjugiert-komplexe Polpaar λ_{ij} und $\lambda_{ij}^\star = \lambda_{ij^\star}$ lassen sich nach [Büh20, S. 31 f.] durch eine bijektive Zustandstransformation stets reellwertige Ersatzzustände $\boldsymbol{\zeta}_1(t)$ und $\boldsymbol{\zeta}_2(t)$ mit reeller Dynamik definieren, die den Sollzustandsverlauf nicht beeinflussen. Durch Einsetzen von $\lambda_{ij}^\star = \lambda_{ij^\star}$ in (4.9) ist ersichtlich, dass $\boldsymbol{D}_{ij^\star} = \boldsymbol{D}_{ij}^\star$ gilt (vgl. (4.14)). Zudem folgt aus (4.7) unter Annahme 4.1 mit (4.12) $\boldsymbol{\zeta}_{ij^\star}(t) = \boldsymbol{\zeta}_{ij}^\star(t)$.

Definiere

$$\boldsymbol{\zeta}_1(t) \coloneqq \boldsymbol{\zeta}_{ij}(t) + \boldsymbol{\zeta}_{ij^\star}(t) = 2\,\mathrm{Re}\{\boldsymbol{\zeta}_{ij}(t)\} \tag{B.10}$$

und

$$\boldsymbol{\zeta}_2(t) \coloneqq \mathrm{j}\left(\boldsymbol{\zeta}_{ij}(t) - \boldsymbol{\zeta}_{ij^\star}(t)\right) = 2\,\mathrm{Im}\{\boldsymbol{\zeta}_{ij}(t)\}. \tag{B.11}$$

Da durch $\boldsymbol{\zeta}_{ij}(t) = \frac{\boldsymbol{\zeta}_1(t) - \mathrm{j}\boldsymbol{\zeta}_2(t)}{2}$ und $\boldsymbol{\zeta}_{ij^\star}(t) = \frac{\boldsymbol{\zeta}_1(t) + \mathrm{j}\boldsymbol{\zeta}_2(t)}{2}$ Umkehrfunktionen existieren, gewährleistet eine Ersetzung von $\boldsymbol{\zeta}_{ij}(t)$ und $\boldsymbol{\zeta}_{ij^\star}(t)$ durch $\boldsymbol{\zeta}_1(t)$ und $\boldsymbol{\zeta}_2(t)$ aufgrund von Proposition 4.2 weiterhin eine eineindeutige Repräsentation der Parameter $\boldsymbol{c}$. Die reellwertigen Zustände $\boldsymbol{\zeta}_1(t)$ und $\boldsymbol{\zeta}_2(t)$ sind durch die reelle Dynamik

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\boldsymbol{\zeta}_1(t)\\ \boldsymbol{\zeta}_2(t)\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{\zeta}}_{ij}(t) + \dot{\boldsymbol{\zeta}}_{ij^\star}(t)\\ \mathrm{j}\left(\dot{\boldsymbol{\zeta}}_{ij}(t) - \dot{\boldsymbol{\zeta}}_{ij^\star}(t)\right)\end{bmatrix} = \begin{bmatrix}\boldsymbol{D}_{ij}\boldsymbol{\zeta}_{ij}(t) + \boldsymbol{D}_{ij}^\star\boldsymbol{\zeta}_{ij}^\star(t)\\ \mathrm{j}\left(\boldsymbol{D}_{ij}\boldsymbol{\zeta}_{ij}(t) - \boldsymbol{D}_{ij}^\star\boldsymbol{\zeta}_{ij}^\star(t)\right)\end{bmatrix} \tag{B.12}$$

$$= \begin{bmatrix}\boldsymbol{D}_{ij}\left(\frac{\boldsymbol{\zeta}_1(t) - \mathrm{j}\boldsymbol{\zeta}_2(t)}{2}\right) + \boldsymbol{D}_{ij}^\star\left(\frac{\boldsymbol{\zeta}_1(t) + \mathrm{j}\boldsymbol{\zeta}_2(t)}{2}\right)\\ \mathrm{j}\left(\boldsymbol{D}_{ij}\left(\frac{\boldsymbol{\zeta}_1(t) - \mathrm{j}\boldsymbol{\zeta}_2(t)}{2}\right) - \boldsymbol{D}_{ij}^\star\left(\frac{\boldsymbol{\zeta}_1(t) + \mathrm{j}\boldsymbol{\zeta}_2(t)}{2}\right)\right)\end{bmatrix} \tag{B.13}$$

$$= \begin{bmatrix}\frac{\boldsymbol{D}_{ij} + \boldsymbol{D}_{ij}^\star}{2} & \mathrm{j}\frac{\boldsymbol{D}_{ij}^\star - \boldsymbol{D}_{ij}}{2}\\ \mathrm{j}\frac{\boldsymbol{D}_{ij} - \boldsymbol{D}_{ij}^\star}{2} & \frac{\boldsymbol{D}_{ij} + \boldsymbol{D}_{ij}^\star}{2}\end{bmatrix}\begin{bmatrix}\boldsymbol{\zeta}_1(t)\\ \boldsymbol{\zeta}_2(t)\end{bmatrix} \tag{B.14}$$

verknüpft. Schließlich müssen noch $\boldsymbol{s}_{ij}$ und $\boldsymbol{s}_{ij^\star}$ (vgl. (4.18)) durch $\boldsymbol{s}_1 = \boldsymbol{s}_{ij}$ und $\boldsymbol{s}_2 = \boldsymbol{0}$ ersetzt werden, damit

$$\boldsymbol{s}_1^\mathsf{T}\boldsymbol{\zeta}_1(t) + \boldsymbol{s}_2^\mathsf{T}\boldsymbol{\zeta}_2(t) = \boldsymbol{\zeta}_{ij}(t) + \boldsymbol{\zeta}_{ij}^\star(t) = \boldsymbol{s}_{ij}^\mathsf{T}\boldsymbol{\zeta}_{ij}(t) + \boldsymbol{s}_{ij^\star}^\mathsf{T}\boldsymbol{\zeta}_{ij^\star}(t) \tag{B.15}$$

gilt und somit $\boldsymbol{x}_{\mathrm{r},i}(t)$ nicht verändert wird.

B.4 Äquivalenz zwischen Problem 4.1 und Problem 4.2

Lemma B.1 (vgl. [Büh20, S. 33 f.])
Problem 4.1 und Problem 4.2 sind äquivalent.

Beweis:
Unter Annahme eines linearen Regelgesetzes der Form $\boldsymbol{\mu}(\tilde{\boldsymbol{x}}(\tau)) = -\tilde{\boldsymbol{K}}\tilde{\boldsymbol{x}}(\tau)$ ergibt sich der Verlauf des erweiterten Zustands (vgl. (4.33)) nach [Föl16, S. 277] zu

$$\tilde{\boldsymbol{x}}(\tau) = \mathrm{e}^{\left(\tilde{\boldsymbol{A}}-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}\right)(\tau-t)}\tilde{\boldsymbol{x}}(t). \tag{B.16}$$

Einsetzen in das Gütefunktional (4.34) ergibt

$$\int_t^\infty e^{-\gamma(\tau-t)}\left(\tilde{\boldsymbol{x}}^\mathsf{T}(\tau)\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{x}}(\tau) + \boldsymbol{\mu}^\mathsf{T}(\boldsymbol{x}(\tau),\boldsymbol{\zeta}(\tau))\boldsymbol{R}\boldsymbol{\mu}(\boldsymbol{x}(\tau),\boldsymbol{\zeta}(\tau))\right)\mathrm{d}\tau \tag{B.17}$$

$$= \tilde{\boldsymbol{x}}^\mathsf{T}(t)\int_t^\infty \mathrm{e}^{-\frac{\gamma}{2}(\tau-t)}\mathrm{e}^{\left(\tilde{\boldsymbol{A}}-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}\right)^\mathsf{T}(\tau-t)}\left(\tilde{\boldsymbol{Q}} + \tilde{\boldsymbol{K}}^\mathsf{T}\boldsymbol{R}\tilde{\boldsymbol{K}}\right)\mathrm{e}^{\left(\tilde{\boldsymbol{A}}-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}\right)(\tau-t)}\mathrm{e}^{-\frac{\gamma}{2}(\tau-t)}\,\mathrm{d}\tau\,\tilde{\boldsymbol{x}}(t) \tag{B.18}$$

$$= \tilde{\boldsymbol{x}}^\mathsf{T}(t)\int_t^\infty \mathrm{e}^{\left(\left(\tilde{\boldsymbol{A}}-\frac{\gamma}{2}\right)-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}\right)^\mathsf{T}(\tau-t)}\left(\tilde{\boldsymbol{Q}} + \tilde{\boldsymbol{K}}^\mathsf{T}\boldsymbol{R}\tilde{\boldsymbol{K}}\right)\mathrm{e}^{\left(\left(\tilde{\boldsymbol{A}}-\frac{\gamma}{2}\right)-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}\right)(\tau-t)}\,\mathrm{d}\tau\,\tilde{\boldsymbol{x}}(t) \tag{B.19}$$

$$= \int_t^\infty \left(\mathrm{e}^{\left(\left(\tilde{\boldsymbol{A}}-\frac{\gamma}{2}\right)-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}\right)^\mathsf{T}(\tau-t)}\tilde{\boldsymbol{x}}(t)\right)^\mathsf{T}\left(\tilde{\boldsymbol{Q}} + \tilde{\boldsymbol{K}}^\mathsf{T}\boldsymbol{R}\tilde{\boldsymbol{K}}\right)\mathrm{e}^{\left(\left(\tilde{\boldsymbol{A}}-\frac{\gamma}{2}\right)-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}\right)(\tau-t)}\tilde{\boldsymbol{x}}(t)\,\mathrm{d}\tau. \tag{B.20}$$

In (B.18) wurden dabei die Zusammenhänge

$$\mathrm{e}^{-\frac{\gamma}{2}(\tau-t)}\mathrm{e}^{\left(\tilde{\boldsymbol{A}}-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}\right)^\mathsf{T}(\tau-t)} = \mathrm{e}^{\left(\tilde{\boldsymbol{A}}-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}-\frac{\gamma}{2}\boldsymbol{I}\right)^\mathsf{T}(\tau-t)} \tag{B.21}$$

und

$$\mathrm{e}^{\left(\tilde{\boldsymbol{A}}-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}\right)(\tau-t)}\mathrm{e}^{-\frac{\gamma}{2}(\tau-t)} = \mathrm{e}^{\left(\tilde{\boldsymbol{A}}-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}-\frac{\gamma}{2}\boldsymbol{I}\right)(\tau-t)} \tag{B.22}$$

genutzt. Der Ausdruck $\mathrm{e}^{\left(\left(\tilde{\boldsymbol{A}}-\frac{\gamma}{2}\right)-\tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}\right)^\mathsf{T}(\tau-t)}\tilde{\boldsymbol{x}}(t)$ in (B.20) entspricht dem Verlauf des erweiterten Zustands für die Systemdynamik $\dot{\tilde{\boldsymbol{x}}}(\tau) = \left(\tilde{\boldsymbol{A}} - \frac{\gamma}{2}\boldsymbol{I}\right)\tilde{\boldsymbol{x}}(\tau) + \tilde{\boldsymbol{B}}\boldsymbol{\mu}(\boldsymbol{x}(\tau),\boldsymbol{\zeta}(\tau))$ für den durch $\boldsymbol{\mu}(\tilde{\boldsymbol{x}}(\tau)) = -\tilde{\boldsymbol{K}}\tilde{\boldsymbol{x}}(\tau)$ geschlossenen Regelkreis. Somit folgt Äquivalenz zwischen Problem 4.1 und Problem 4.2. Zudem rechtfertigt die LQ-Form von Problem 4.2 die Annahme eines linearen Regelgesetzes (vgl. [Büh20, S. 33 f.], [ML14a, Beweis zu Lemma 2]). □

B.5 Beweis zu Lemma 4.3

Beweis:
Es gilt

$$\mathrm{Rang}(\boldsymbol{M}) = \mathrm{Dim}(\mathrm{Bild}(\boldsymbol{M})) = n_1 + n_2 \tag{B.23}$$

$$\Leftrightarrow \forall \boldsymbol{z} = \begin{bmatrix}\boldsymbol{z}_1\\ \boldsymbol{z}_2\end{bmatrix} \in \mathbb{R}^{n_1+n_2}\ \exists \boldsymbol{v} = \begin{bmatrix}\boldsymbol{v}_1\\ \boldsymbol{v}_2\end{bmatrix} : \boldsymbol{M}\boldsymbol{v} = \boldsymbol{z}. \tag{B.24}$$

Sei ein solches beliebiges $\boldsymbol{z}$ gegeben. Aus

$$\begin{bmatrix} \boldsymbol{M}_1 & \boldsymbol{M}_3 \\ \boldsymbol{0} & \boldsymbol{M}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix} \tag{B.25}$$

folgt

$$\boldsymbol{M}_2 \boldsymbol{v}_2 = \boldsymbol{z}_2. \tag{B.26}$$

Aufgrund von (4.53) existiert $\boldsymbol{v}_2$. Des Weiteren gilt

$$\boldsymbol{M}_1 \boldsymbol{v}_1 = \boldsymbol{z}_1 - \boldsymbol{M}_3 \boldsymbol{v}_2. \tag{B.27}$$

Wegen (4.52) existiert auch $\boldsymbol{v}_1$. Daher existiert $\boldsymbol{v}$, es folgt (B.24) und somit gilt (4.55). □

B.6 Beweis zu Lemma 4.4

Beweis:
Mit dem Hautus-Kriterium für Stabilisierbarkeit [ZD98, Theorem 3.2] folgt, dass $\left(\tilde{\boldsymbol{A}}', \tilde{\boldsymbol{B}}\right)$ genau dann stabilisierbar ist, wenn $\left[\tilde{\boldsymbol{A}}' - \lambda \boldsymbol{I} \quad \tilde{\boldsymbol{B}}\right]$ für alle nicht-negativen Eigenwerte λ von $\tilde{\boldsymbol{A}}'$ Maximalrang aufweist. Es gilt

$$\begin{aligned} \operatorname{Rang}\left[\tilde{\boldsymbol{A}}' - \lambda \boldsymbol{I} \quad \tilde{\boldsymbol{B}}\right] &= \operatorname{Rang}\begin{bmatrix} \boldsymbol{A} - \lambda \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{B} \\ \boldsymbol{0} & \left(\boldsymbol{D} - \frac{\gamma}{2}\boldsymbol{I}\right) - \lambda \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \\ &= \operatorname{Rang}\left[\begin{array}{cc:c} \boldsymbol{A} - \lambda \boldsymbol{I} & \boldsymbol{B} & \boldsymbol{0} \\ \hdashline \boldsymbol{0} & \boldsymbol{0} & \left(\boldsymbol{D} - \frac{\gamma}{2}\boldsymbol{I}\right) - \lambda \boldsymbol{I} \end{array}\right]. \end{aligned} \tag{B.28}$$

Da $(\boldsymbol{A}, \boldsymbol{B})$ aufgrund von Annahme 4.2 stabilisierbar ist, hat nach dem Hautus-Kriterium $\left[\boldsymbol{A} - \lambda \boldsymbol{I} \quad \boldsymbol{B}\right]$ vollen Rang. Zudem ist $\left(\boldsymbol{D} - \frac{\gamma}{2}\boldsymbol{I}\right) - \lambda \boldsymbol{I}$ regulär, da $\boldsymbol{D} - \frac{\gamma}{2}\boldsymbol{I}$ für $\gamma > \gamma_{\text{min}}$ ausschließlich negative Eigenwerte besitzt. Nach Lemma 4.3 weist der Ausdruck in (B.28) somit vollen Rang auf. Somit sind alle instabilen Eigenwerte steuerbar, woraus Stabilisierbarkeit von $\left(\tilde{\boldsymbol{A}}', \tilde{\boldsymbol{B}}\right)$ resultiert. □

B.7 Beweis zu Lemma 4.6

Beweis:
Mit dem Hautus-Kriterium für Detektierbarkeit [ZD98, Theorem 3.4] folgt, dass $\left(\tilde{\boldsymbol{A}}', \sqrt{\tilde{\boldsymbol{Q}}}\right)$

genau dann detektierbar ist, wenn $\begin{bmatrix} \tilde{\boldsymbol{A}}' - \lambda \boldsymbol{I} \\ \sqrt{\tilde{\boldsymbol{Q}}} \end{bmatrix}$ für alle nicht-negativen Eigenwerte λ von $\tilde{\boldsymbol{A}}'$ vollen Rang aufweist. Aus

$$\tilde{\boldsymbol{Q}} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{S} \end{bmatrix}^{\mathsf{T}} \boldsymbol{Q} \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{S} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{S} \end{bmatrix}^{\mathsf{T}} \sqrt{\boldsymbol{Q}}^{\mathsf{T}} \sqrt{\boldsymbol{Q}} \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{S} \end{bmatrix} \tag{B.29}$$

folgt mit der Definition von $\tilde{\boldsymbol{Q}}$ nach (4.36)

$$\sqrt{\tilde{\boldsymbol{Q}}} = \sqrt{\boldsymbol{Q}} \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{S} \end{bmatrix}. \tag{B.30}$$

Daher gilt

$$\begin{aligned} \operatorname{Rang} \begin{bmatrix} \tilde{\boldsymbol{A}}' - \lambda \boldsymbol{I} \\ \sqrt{\tilde{\boldsymbol{Q}}} \end{bmatrix} &= \operatorname{Rang} \begin{bmatrix} \boldsymbol{A} - \lambda \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \left(\boldsymbol{D} - \frac{\gamma}{2}\boldsymbol{I}\right) - \lambda \boldsymbol{I} \\ \sqrt{\boldsymbol{Q}} & -\sqrt{\boldsymbol{Q}}\boldsymbol{S} \end{bmatrix} \\ &= \operatorname{Rang} \left[\begin{array}{c:c} \boldsymbol{A} - \lambda \boldsymbol{I} & \boldsymbol{0} \\ \sqrt{\boldsymbol{Q}} & -\sqrt{\boldsymbol{Q}}\boldsymbol{S} \\ \hdashline \boldsymbol{0} & \left(\boldsymbol{D} - \frac{\gamma}{2}\boldsymbol{I}\right) - \lambda \boldsymbol{I} \end{array} \right]^{\mathsf{T}}. \end{aligned} \tag{B.31}$$

Unter Annahme 4.2 ist $(\boldsymbol{A}, \sqrt{\boldsymbol{Q}})$ detektierbar, daher besitzt $\begin{bmatrix} \boldsymbol{A} - \lambda \boldsymbol{I} \\ \sqrt{\boldsymbol{Q}} \end{bmatrix}$ nach dem Hautus-Kriterium vollen Rang. Zudem ist $\left(\boldsymbol{D} - \frac{\gamma}{2}\boldsymbol{I}\right) - \lambda \boldsymbol{I}$ regulär, da $\boldsymbol{D} - \frac{\gamma}{2}\boldsymbol{I}$ für $\gamma > \gamma_{\text{min}}$ ausschließlich negative Eigenwerte besitzt. Mit Lemma 4.5 folgt somit, dass (B.31) Maximalrang aufweist. Daher sind alle nicht-negativen Eigenwerte beobachtbar und $\left(\tilde{\boldsymbol{A}}', \sqrt{\tilde{\boldsymbol{Q}}}\right)$ ist detektierbar. □

B.8 Value Iteration nach Bian und Jiang [BJ16a]

Einsetzen von (4.73) und (4.74) in (4.72) ergibt

$$\frac{\partial V^{\boldsymbol{\mu}_s}(\tilde{\boldsymbol{x}})}{\partial s} = \boldsymbol{\phi}^{\mathsf{T}}(\tilde{\boldsymbol{x}}) \frac{\mathrm{d}\hat{\boldsymbol{w}}_\phi(s)}{\mathrm{d}s} = \min_{\boldsymbol{\mu}} \left(r(\tilde{\boldsymbol{x}}, \boldsymbol{\mu}) + \boldsymbol{\psi}^{\mathsf{T}}(\tilde{\boldsymbol{x}}, \boldsymbol{\mu}) \hat{\boldsymbol{w}}_\psi(s) \right). \tag{B.32}$$

Analog zu [BJ16a] wird ein Datensatz aus $M \geq \frac{\tilde{n}(\tilde{n}+1)}{2} + \tilde{n}p$ Tupeln $\left(\tilde{\boldsymbol{x}}(t_j), \boldsymbol{\Theta}_j, \tilde{\boldsymbol{x}}(t_j + T_{\text{IRL}})\right)$, $j = 1, \ldots, M$, verwendet, wobei

$$\boldsymbol{\Theta}_j := \int_{t_j}^{t_j + T_{\text{IRL}}} \boldsymbol{\psi}(\tilde{\boldsymbol{x}}, \boldsymbol{\mu}) \, \mathrm{d}\tau \tag{B.33}$$

definiert ist. Mithilfe der Messdatenmatrizen

$$\boldsymbol{D}_\phi := \frac{1}{M} \sum_{j=1}^{M} \boldsymbol{\phi}(\tilde{\boldsymbol{x}}_j) \boldsymbol{\phi}^\intercal(\tilde{\boldsymbol{x}}_j), \tag{B.34}$$

$$\boldsymbol{D}_\psi := \frac{1}{M} \sum_{j=1}^{M} \boldsymbol{\Theta}_j \boldsymbol{\Theta}_j^\intercal \tag{B.35}$$

folgt, da die skalare Gleichung (B.32) für jedes der M Tupel gilt, nach Multiplikation von $\boldsymbol{\phi}(\tilde{\boldsymbol{x}}_j)$ von links und Summation über alle M Datentupel

$$\sum_{j=1}^{M} \boldsymbol{\phi}(\tilde{\boldsymbol{x}}_j) \boldsymbol{\phi}^\intercal(\tilde{\boldsymbol{x}}_j) \frac{\mathrm{d}\hat{\boldsymbol{w}}_\phi(s)}{\mathrm{d}s} = \sum_{j=1}^{M} \boldsymbol{\phi}(\tilde{\boldsymbol{x}}_j) \min_{\boldsymbol{\mu}} \left(r(\tilde{\boldsymbol{x}}_j, \boldsymbol{\mu}) + \boldsymbol{\psi}^\intercal(\tilde{\boldsymbol{x}}_j, \boldsymbol{\mu}) \hat{\boldsymbol{w}}_\psi(s) \right). \tag{B.36}$$

Falls $\boldsymbol{D}_\phi$ invertierbar ist, resultiert somit die Adaptionsvorschrift

$$\frac{\mathrm{d}\hat{\boldsymbol{w}}_\phi(s)}{\mathrm{d}s} = \frac{1}{M} \boldsymbol{D}_\phi^{-1} \sum_{j=1}^{M} \boldsymbol{\phi}(\tilde{\boldsymbol{x}}_j) \min_{\boldsymbol{\mu}} \big(r(\tilde{\boldsymbol{x}}_j, \boldsymbol{\mu}) + \boldsymbol{\psi}^\intercal(\tilde{\boldsymbol{x}}_j, \boldsymbol{\mu}) \hat{\boldsymbol{w}}_\psi(s) \big) \tag{B.37}$$

der Gewichte $\hat{\boldsymbol{w}}_\phi(s)$. Aus $\hat{\boldsymbol{w}}_\phi(s)$ lässt sich, falls $\boldsymbol{D}_\psi$ invertierbar ist, wegen[146]

$$\boldsymbol{\Theta}_j^\intercal \hat{\boldsymbol{w}}_\psi(s) = \Big(\boldsymbol{\phi}\big(\tilde{\boldsymbol{x}}(t_j + T_{\mathrm{IRL}})\big) - \boldsymbol{\phi}\big(\tilde{\boldsymbol{x}}(t_j)\big) \Big)^\intercal \hat{\boldsymbol{w}}_\phi(s) \tag{B.38}$$

direkt

$$\hat{\boldsymbol{w}}_\psi(s) = \boldsymbol{\Theta} \hat{\boldsymbol{w}}_\phi(s) \tag{B.39}$$

mit

$$\boldsymbol{\Theta} := \frac{1}{M} \boldsymbol{D}_\psi^{-1} \sum_{j=1}^{M} \Big(\boldsymbol{\phi}\big(\tilde{\boldsymbol{x}}(t_j + T_{\mathrm{IRL}})\big) - \boldsymbol{\phi}\big(\tilde{\boldsymbol{x}}(t_j)\big) \Big)^\intercal \tag{B.40}$$

berechnen. Die Konvergenz dieser Value Iteration beginnend mit einer positiv definiten initialen Value Function wird in [BJ16a, Theorem 2] untersucht.

[146] Gleichung (B.38) folgt direkt aus $V^{\boldsymbol{\mu}_s}(\tilde{\boldsymbol{x}}(t_j + T_{\mathrm{IRL}})) - V^{\boldsymbol{\mu}_s}(\tilde{\boldsymbol{x}}(t_j)) = \int_{t_j}^{t_j + T_{\mathrm{IRL}}} \nabla_{\tilde{\boldsymbol{x}}} V^{\boldsymbol{\mu}_s}(\tilde{\boldsymbol{x}}) \frac{\mathrm{d}\tilde{\boldsymbol{x}}}{\mathrm{d}\tau} \, \mathrm{d}\tau$.

C Anhang zu Kapitel 5

C.1 Beweisskizze zu Lemma 5.2

Beweis:
Einsetzen eines Elements $x_{\mathrm{PE},o}$ von $\boldsymbol{x}_{\mathrm{PE}}$ (5.30) in einen der Faktoren $x_o^{f_{o,\bar{h}}}$ von $\phi_{\bar{h}}(\boldsymbol{x})$ (5.25) und Anwendung des Multinomialtheorems [Spi68, S. 4] liefert

$$\begin{aligned} x_{\mathrm{PE},o}^{f_{o,\bar{h}}} &= \left(\sum_{j=1}^{m} g_{j,o}\sin(\omega_j t) + \sum_{j=1}^{m} \bar{g}_{j,o}\cos(\omega_j t)\right)^{f_{o,\bar{h}}} \\ &= \sum_{r_1+\dots+r_{2m}=f_{o,\bar{h}}} \frac{f_{o,\bar{h}}!}{r_1!\dots r_{2m}!} g_{1,o}^{r_1}\sin^{r_1}(\omega_1 t)\dots\bar{g}_{m,o}^{r_{2m}}\cos^{r_{2m}}(\omega_m t). \end{aligned} \tag{C.1}$$

Unter der beispielhaften Annahme, dass r_1 geradzahlig ist, ergibt sich aus der Potenzfunktion[147] von $\sin^{r_1}(\omega_1 t)$ [Spi68, S. 17] aus (C.1)

$$\begin{aligned} x_{\mathrm{PE},o}^{f_{o,\bar{h}}} = \sum_{r_1+\dots+r_{2m}=f_{o,\bar{h}}} &\frac{f_{o,\bar{h}}!}{r_1!\dots r_{2m}!} g_{1,o}^{r_1}\dots\bar{g}_{m,o}^{r_{2m}} \\ &\left(\frac{1}{2^{r_1}}\binom{r_1}{\frac{r_1}{2}} + \frac{1}{2^{r_1-1}}\sum_{s=0}^{\frac{r_1}{2}-1}(-1)^{\frac{r_1}{2}-s}\binom{r_1}{s}\cos\Big((r_1-2s)\omega_1 t\Big)\right) \\ &\dots\cos^{r_{2m}}(\omega_m t). \end{aligned} \tag{C.2}$$

Da für den ungeradzahligen Fall sowie die Kosinusfunktionen ähnliche Potenzfunktionen existieren (vgl. [Spi68, S. 17]), kann (C.2) unter zusätzlicher Anwendung der Produktregeln[148] für $\sin(\cdot)\sin(\cdot)$, $\sin(\cdot)\cos(\cdot)$ und $\cos(\cdot)\cos(\cdot)$ zu

$$x_{\mathrm{PE},o}^{f_{o,\bar{h}}} = \sum_{l=1}^{L_{\bar{h}}^{\langle o\rangle}} a_{l,\bar{h}}^{\langle o\rangle}\sin\left(\sum_{j=1}^{m} b_{j,l,\bar{h}}^{\langle o\rangle}\omega_j t\right) + \sum_{k=1}^{K_{\bar{h}}^{\langle o\rangle}} c_{k,\bar{h}}^{\langle o\rangle}\cos\left(\sum_{j=1}^{m} d_{j,k,\bar{h}}^{\langle o\rangle}\omega_j t\right) + e_{\bar{h}}^{\langle o\rangle} \tag{C.3}$$

[147] Auch bekannt als *power reduction formula.*
[148] Bekannt als *product-to-sum identities.*

umgeformt werden. Die Indizes $\bar{h}$ und $\langle o\rangle$, mit denen die Parameter $a_{l,\bar{h}}^{\langle o\rangle}, c_{k,\bar{h}}^{\langle o\rangle} \in \mathbb{R}_{\neq 0}$, $e_{\bar{h}}^{\langle o\rangle}, b_{j,l,\bar{h}}^{\langle o\rangle}, d_{j,k,\bar{h}}^{\langle o\rangle} \in \mathbb{R}$ in (C.3) gekennzeichnet sind, indizieren die Abhängigkeit von den Elementen $\phi_{\bar{h}}(\boldsymbol{x}_{\mathrm{PE}})$ bzw. dem Faktor $x_{\mathrm{PE},o}^{f_{o,\bar{h}}}$. Um $a_{l,\bar{h}}^{\langle o\rangle}, c_{k,\bar{h}}^{\langle o\rangle} \neq 0$ annehmen zu können, werden die oberen Summengrenzen $L_{\bar{h}}^{\langle o\rangle}, K_{\bar{h}}^{\langle o\rangle} \in \mathbb{N}_{\geq 0}$ mit denselben Indizes versehen.

Da (C.3) $\forall o \in \{1,\ldots,n\}$ gilt, folgt für jedes Element $\phi_{\bar{h}}(\boldsymbol{x}_{\mathrm{PE}})$ von $\boldsymbol{\phi}(\boldsymbol{x}_{\mathrm{PE}})$

$$\begin{aligned}\phi_{\bar{h}}(\boldsymbol{x}_{\mathrm{PE}}) &= \bar{f}_{\bar{h}} \prod_{o=1}^{n} x_{\mathrm{PE},o}^{f_{o,\bar{h}}} \\ &= \sum_{l=1}^{L_{\bar{h}}} a_{l,\bar{h}} \sin\left(\sum_{j=1}^{m} b_{j,l,\bar{h}}\omega_j t\right) + \sum_{k=1}^{K_{\bar{h}}} c_{k,\bar{h}} \cos\left(\sum_{j=1}^{m} d_{j,k,\bar{h}}\omega_j t\right) + e_{\bar{h}}.\end{aligned} \tag{C.4}$$

Die Umformung in (C.4) ergibt sich aus der Ausmultiplikation des Produktes der Summen (C.3) und erneuter Anwendung der Produktregeln für $\sin(\cdot)\sin(\cdot)$, $\sin(\cdot)\cos(\cdot)$ und $\cos(\cdot)\cos(\cdot)$. Aufgrund von $f_{o,\bar{h}} \neq 0$ für mindestens ein o (vgl. Annahme 5.3) sowie $m \geq 1$ und $g_{j,o} \neq 0$ oder $\bar{g}_{j,o} \neq 0$ für mindestens ein j (vgl. Annahme 5.4) folgt, dass die Summenobergrenzen $L_{\bar{h}} + K_{\bar{h}} \geq 1, \forall \bar{h} \in \{1,\ldots,h\}$, erfüllen. Mit der Kurzschreibweise $\underline{\omega}_{l,\bar{h}} := \sum_{j=1}^{m} b_{j,l,\bar{h}}\omega_j$ und $\bar{\underline{\omega}}_{k,\bar{h}} := \sum_{j=1}^{m} d_{j,k,\bar{h}}\omega_j$ folgt aus (C.4) schließlich (5.31). □

C.2 Beweis zu Lemma 5.9

Beweis:
Einsetzen von (5.53) in (5.25) liefert $\forall \bar{h} \in \{1,\ldots,h\}$

$$\phi_{\bar{h}}(\bar{\boldsymbol{x}}_{\mathrm{PE}}) = \bar{f}_{\bar{h}} \prod_{j=1}^{n} \bar{x}_{\mathrm{PE},j}^{f_{j,\bar{h}}} = \left(\prod_{j=1}^{n} \nu_j^{f_{j,\bar{h}}}\right)\left(\bar{f}_{\bar{h}} \prod_{j=1}^{n} x_{\mathrm{PE},j}^{f_{j,\bar{h}}}\right) = \left(\prod_{j=1}^{n} \nu_j^{f_{j,\bar{h}}}\right)\phi_{\bar{h}}(\boldsymbol{x}_{\mathrm{PE}}). \tag{C.5}$$

Nach (C.5) beeinflusst eine Skalierung von $\boldsymbol{x}_{\mathrm{PE}}$ wie in (5.53) lediglich die Koeffizienten $a_{l,\bar{h}}$, $c_{k,\bar{h}}$ und $e_{\bar{h}}$ in (5.31). Da die exakten Werte dieser Koeffizienten jedoch irrelevant für die Berechnung der Menge Ω sowie die Erfüllung von (5.35) sind, gewährleistet jeder Vektor $\boldsymbol{\omega} \in \Omega$, dass $\bar{\boldsymbol{x}}_{\mathrm{PE}}$ SR bezüglich $\dot{\boldsymbol{\phi}}(\cdot)$ ist, wenn Ω basierend auf Satz 5.2 mit $\boldsymbol{x}_{\mathrm{PE}}$ berechnet wurde. □

C.3 $\boldsymbol{M}_i$ und $\boldsymbol{v}_{\mathrm{freq}}(t)$ für das verwendete Beispielsystem

Im Folgenden wird gezeigt, dass das in Abschnitt 5.7 betrachtete Beispielsystem die durch (5.79) gegebene Bedingung in Annahme 5.6 sowie $\mathrm{Rang}(\boldsymbol{M}_i) = h_i$ erfüllt. Da $\check{x}_1$, mit $\check{\boldsymbol{x}}$ wie

in (5.78), einem flachen Ausgang $h(\check{\boldsymbol{x}}) = \check{x}_1$ des Systems (5.85) entspricht, folgt für die Wahl von $y_{\text{f}}|_{\bar{\boldsymbol{x}}_{\text{PE}}} = h_{\text{f}}\left(\boldsymbol{z}_{\text{B}}|_{\bar{\boldsymbol{x}}_{\text{PE}}}\right) = h_{\text{f}}(\boldsymbol{t}(\bar{\boldsymbol{x}}_{\text{PE}})) = \bar{x}_{\text{PE},1}$ mit (5.66) und (5.88)

$$\check{x}_1 = \bar{x}_{\text{PE},1} = \nu_1\big(\sin(\omega_{\text{c},1}t) + \sin(\omega_{\text{c},2}t)\big). \tag{C.6}$$

Nach (5.72) gilt $z_{\text{B},2} = \dot{z}_{\text{B},1}$ und aus (5.74) folgt $z_{\text{B},1} = t_t(\check{\boldsymbol{x}}) = h(\check{\boldsymbol{x}}) = \check{x}_1$. Daraus ergibt sich

$$z_{\text{B},2} = \dot{z}_{\text{B},1} = \frac{\text{d}}{\text{d}t}\check{x}_1 = \nu_1\big(\omega_{\text{c},1}\cos(\omega_{\text{c},1}t) + \omega_{\text{c},2}\cos(\omega_{\text{c},2}t)\big). \tag{C.7}$$

Aufgrund von

$$z_{\text{B},2} = L_{\boldsymbol{f}_{\text{g}}}h(\check{\boldsymbol{x}}) = -2\check{x}_1 + \check{x}_2 \tag{C.8}$$

(vgl. (5.74)) folgt mit (C.7)

$$\begin{aligned}\check{x}_2 &= 2\check{x}_1 + z_{\text{B},2} = 2\bar{x}_{\text{PE},1} + z_{\text{B},2}\\ &= \nu_1\big(2\sin(\omega_{\text{c},1}t) + 2\sin(\omega_{\text{c},2}t) + \omega_{\text{c},1}\cos(\omega_{\text{c},1}t) + \omega_{\text{c},2}\cos(\omega_{\text{c},2}t)\big).\end{aligned} \tag{C.9}$$

Wie (C.6) und (C.9) verdeutlichen, gilt für das gewählte Beispiel $\check{\boldsymbol{x}} \neq \bar{\boldsymbol{x}}_{\text{PE}}$. Um zu zeigen, dass (5.79) dennoch gilt, wird $\check{\boldsymbol{\sigma}}_i$ berechnet. Es folgt für $i \in \mathcal{N}$

$$\begin{aligned}\check{\boldsymbol{\sigma}}_i &= \dot{\boldsymbol{\phi}}_i(\check{\boldsymbol{x}})\\ &= \nu_1^2 \underbrace{\begin{bmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 & b_4 & 0\\ b_5 & a_1 & b_6 & a_2 & b_7 & a_3 & b_8 & a_4\\ b_9 & a_5 & b_{10} & a_6 & b_{11} & a_7 & b_{12} & a_8 \end{bmatrix}}_{=:\boldsymbol{M}_i} \underbrace{\begin{bmatrix} \sin\big((\omega_{\text{c},1}-\omega_{\text{c},2})t\big)\\ \cos\big((\omega_{\text{c},1}-\omega_{\text{c},2})t\big)\\ \sin\big((\omega_{\text{c},1}+\omega_{\text{c},2})t\big)\\ \cos\big((\omega_{\text{c},1}+\omega_{\text{c},2})t\big)\\ \sin(2\omega_{\text{c},1}t)\\ \cos(2\omega_{\text{c},1}t)\\ \sin(2\omega_{\text{c},2}t)\\ \cos(2\omega_{\text{c},2}t) \end{bmatrix}}_{=:\frac{\text{d}\boldsymbol{v}_{\text{freq}}(t)}{\text{d}t}}\end{aligned} \tag{C.10}$$

mit

$$\begin{aligned}
a_1 &= -\frac{(\omega_{\text{c},1}-\omega_{\text{c},2})^2}{2}, & a_2 &= \frac{(\omega_{\text{c},1}+\omega_{\text{c},2})^2}{2},\\
a_3 &= \omega_{\text{c},1}^2, & a_4 &= \omega_{\text{c},2}^2,\\
a_5 &= -2(\omega_{\text{c},1}-\omega_{\text{c},2})^2, & a_6 &= 2(\omega_{\text{c},1}+\omega_{\text{c},2})^2,\\
a_7 &= 4\omega_{\text{c},1}^2, & a_8 &= 4\omega_{\text{c},2}^2,\\
b_1 &= \omega_{\text{c},2}-\omega_{\text{c},1}, & b_2 &= \omega_{\text{c},1}+\omega_{\text{c},2},\\
b_3 &= \omega_{\text{c},1}, & b_4 &= \omega_{\text{c},2},\\
b_5 &= 2\left(\omega_{\text{c},2}-\omega_{\text{c},1}\right), & b_6 &= 2\left(\omega_{\text{c},1}+\omega_{\text{c},2}\right),\\
b_7 &= 2\omega_{\text{c},1}, & b_8 &= 2\omega_{\text{c},2},\\
b_9 &= \left(\omega_{\text{c},1}\omega_{\text{c},2}+4\right)\left(\omega_{\text{c},2}-\omega_{\text{c},1}\right), & b_{10} &= \left(4-\omega_{\text{c},1}\omega_{\text{c},2}\right)\left(\omega_{\text{c},1}+\omega_{\text{c},2}\right),\\
b_{11} &= \omega_{\text{c},1}\left(4-\omega_{\text{c},1}^2\right), & b_{12} &= \omega_{\text{c},2}\left(4-\omega_{\text{c},2}^2\right).
\end{aligned} \tag{C.11}$$

Weiterhin gilt Rang$(\boldsymbol{M}_i) = h_i = 3$, $\forall \boldsymbol{\omega}_\mathrm{c} \in \Omega$. Schließlich muss noch gezeigt werden, dass die Frequenzen gleichartiger trigonometrischer Funktionen in $\boldsymbol{v}_\mathrm{freq}(t)$ unterschiedlich sind. Dies ist aufgrund von $\omega_{\mathrm{c},1} \neq 0$, $\omega_{\mathrm{c},2} \neq 0$, $\omega_{\mathrm{c},1} \neq \omega_{\mathrm{c},2}$, $\omega_{\mathrm{c},1} \neq -\omega_{\mathrm{c},2}$, $\omega_{\mathrm{c},1} \neq 3\omega_{\mathrm{c},2}$, $\omega_{\mathrm{c},1} \neq -3\omega_{\mathrm{c},2}$, $\omega_{\mathrm{c},2} \neq 3\omega_{\mathrm{c},1}$ und $\omega_{\mathrm{c},2} \neq -3\omega_{\mathrm{c},1}$ für beliebige $\boldsymbol{\omega}_\mathrm{c} \in \Omega$ sichergestellt (vgl. die durch $\boldsymbol{C}_z$ beschriebenen Bedingungen in Tabelle 5.1).

D Anhang zu Kapitel 6

D.1 Exkurs: Reinforcement Learning mit unvollständiger Zustandsinformation

Das Ignorieren nicht gemessener Zustände stellt eine Verletzung der Markov-Annahme dar. Der Lernprozess litte dann unter hoher Varianz und würde langsamer und ungenauer, da der RL-Agent die Value Function im Allgemeinen rein basierend auf der Ausgangsgröße nicht exakt schätzen kann [PRH19], [Wer13, Abschnitt 1.3.1]. Der Umgang mit nur teilweise bekannten Zustandsgrößen stellt somit eine besondere Herausforderung bei lernbasierten Verfahren dar [Rec19], [DAMH19].

Eine Alternative dazu, die nicht messbaren Zustände der Aktuatordynamik zu ignorieren, stellt die Rekonstruktion fehlender Information aus vergangenen Stellgrößen oder Messungen vergangener Ausgangsgrößen dar. Da im betrachteten Anwendungsbeispiel die Längsdynamik jedoch nicht exakt bekannt ist, erweist sich ein modellbasierter Beobachterentwurf (vgl. beispielsweise [CDG93]) als ungeeignet. Grundsätzlich könnte hier die Verwendung eines rekurrenten neuronalen Netzes für die Approximation der Value Function Abhilfe schaffen [HS15]. Die Nutzung linearer rekurrenter Zellen führt bei deren Anwendung in einem realen Fahrzeug jedoch zu sehr langsamer Konvergenz der Actor-Gewichte [PKRH20]. Zudem erschweren rekurrente neuronale Netze die Verwendung von *Experience Replay* [MKS+13], [MKS+15], einem Mechanismus, der Daten mehrfach verwendet und somit zu höherer Dateneffizienz führt (vgl. auch Anhang D.2).

Alternativ kann der Zustandsvektor um vergangene Ausgangs- oder Stellgrößen erweitert werden [PMK99], [MKS+15], um daraus die fehlende Zustandsinformation ohne Verwendung eines Systemmodells zu rekonstruieren. Diese Idee wird auch in der vorliegenden Arbeit in Form eines FIR-Filters zur Schätzung eines Hilfszustands genutzt (vgl. [PRH19], [PKRH20]). Eine ausführlichere Diskussion bezüglich des Umgangs mit nur teilweise messbaren Zuständen ist in den Arbeiten von Puccetti et al. [PRH19] sowie Werbos [Wer13, Abschnitt 1.3.1] zu finden.

D.2 Actor-Critic-Grundlagen

In diesem Abschnitt wird eine kurze Einführung in die in Abschnitt 6.1 verwendeten Actor-Critic-Mechanismen gegeben. Eine ausführlichere Übersicht ist beispielsweise im Grundlagenwerk von Sutton und Barto [SB18] zu finden.

Im Folgenden seien $\boldsymbol{x}_k$ der (gegebenenfalls erweiterte) Zustandsvektor, wie er dem RL-Agenten zur Verfügung gestellt wird[149], $\boldsymbol{u}_k$ die Stellgröße und $r(\boldsymbol{x}_k, \boldsymbol{u}_k)$ die resultierenden Einschrittkosten. Als *Actor* wird das durch $\boldsymbol{\theta}$ parametrierte Regelgesetz $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_k)$ bezeichnet, das zum Ziel hat, die Value Function

$$V^{\boldsymbol{\mu}_{\boldsymbol{\theta}}}(\boldsymbol{x}_k) = \sum_{\kappa=0}^{\infty} \gamma^{\kappa} r\big(\boldsymbol{x}_{k+\kappa}, \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_{k+\kappa})\big) \tag{D.1}$$

zu minimieren. Der *Critic* hingegen sei ein Funktionsapproximator $Q_{\boldsymbol{w}}^{\boldsymbol{\mu}_{\boldsymbol{\theta}}}$ mit dem Parameter $\boldsymbol{w}$, der die Q-Function

$$Q^{\boldsymbol{\mu}_{\boldsymbol{\theta}}}(\boldsymbol{x}_k, \boldsymbol{u}_k) = r(\boldsymbol{x}_k, \boldsymbol{u}_k) + \gamma Q^{\boldsymbol{\mu}_{\boldsymbol{\theta}}}\big(\boldsymbol{x}_{k+1}, \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_{k+1})\big) \tag{D.2}$$

approximiert. Aufgrund der Verwendung der Approximation $Q_{\boldsymbol{w}}^{\boldsymbol{\mu}_{\boldsymbol{\theta}}}$ anstelle von $Q^{\boldsymbol{\mu}_{\boldsymbol{\theta}}}$ ergibt sich der TD-Fehler

$$\delta = r(\boldsymbol{x}_k, \boldsymbol{u}_k) - Q_{\boldsymbol{w}}^{\boldsymbol{\mu}_{\boldsymbol{\theta}}}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \gamma Q_{\boldsymbol{w}}^{\boldsymbol{\mu}_{\boldsymbol{\theta}}}\big(\boldsymbol{x}_{k+1}, \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_{k+1})\big) \tag{D.3}$$

aus (D.2). Mit $L \coloneqq \frac{1}{2}\delta^2$ und dem Gradienten[150]

$$\frac{\partial L}{\partial \boldsymbol{w}} = -\delta \frac{\partial Q_{\boldsymbol{w}}^{\boldsymbol{\mu}_{\boldsymbol{\theta}}}(\boldsymbol{x}_k, \boldsymbol{u}_k)}{\partial \boldsymbol{w}} \tag{D.4}$$

kann der Parametervektor $\boldsymbol{w}$ des Critics mittels Gradientenabstieg oder anderer Optimierungsalgorithmen adaptiert werden.

Die Anpassung des Actor-Gewichts $\boldsymbol{\theta}$ hat hingegen die Minimierung von $Q_{\boldsymbol{w}}^{\boldsymbol{\mu}_{\boldsymbol{\theta}}}$ zum Ziel. Hierfür kann der sogenannte *Deterministic Policy Gradient* (DPG) [SLH$^+$14]

$$\frac{\partial V^{\boldsymbol{\mu}_{\boldsymbol{\theta}}}}{\partial \boldsymbol{\theta}} = \left.\frac{\partial Q_{\boldsymbol{w}}^{\boldsymbol{\mu}_{\boldsymbol{\theta}}}(\boldsymbol{x}_k, \boldsymbol{u}_k)}{\partial \boldsymbol{u}_k}\right|_{\boldsymbol{u}_k = \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_k)}^{\mathsf{T}} \frac{\partial \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_k)}{\partial \boldsymbol{\theta}} \tag{D.5}$$

verwendet werden, um einen Gradientenabstieg oder andere gradientenbasierte Optimierungsmethoden anzuwenden und den Reglerparameter $\boldsymbol{\theta}$ zu adaptieren.

Während grundsätzlich beliebige Stellgrößen $\boldsymbol{u}_k$ auf das System angewandt werden können, approximiert $Q_{\boldsymbol{w}}^{\boldsymbol{\mu}_{\boldsymbol{\theta}}}$, welches nach (D.4) adaptiert wird, die Q-Function, die mit dem durch $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ gegebenen Regelgesetz assoziiert ist. Somit liegt eine Off-Policy-Methode vor (vgl. Abschnitt 2.1.4.4) und Anregungsrauschen führt nicht zu einem Offset der Critic-Schätzung. Ein weiterer, wesentlicher Vorteil der Off-Policy-Charakteristik ist, dass die zum Training verwendeten Datentupel verändert werden können, solange sie konsistent bezüglich der Bellman-Gleichung (D.2) und der zugrunde liegenden Systemdynamik bleiben. Diese Eigenschaft wird in Abschnitt 6.1.2.3 genutzt, indem zusätzliche Anregungssignale auf die Referenzparameter $\boldsymbol{z}_k$ addiert werden. Schließlich kann bei dem verwendeten Off-Policy-Actor-Critic-Verfahren

[149] Im Falle eines erweiterten Zustandsvektors wird im Folgenden somit $\boldsymbol{x}_k$ durch $\tilde{\boldsymbol{x}}_k$ ersetzt.

[150] In dieser Gradientenberechnung wird der Zielwert $Q_{\boldsymbol{w}}^{\boldsymbol{\mu}_{\boldsymbol{\theta}}}\big(\boldsymbol{x}_{k+1}, \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_{k+1})\big)$ üblicherweise als konstant behandelt, siehe beispielsweise [SLH$^+$14, Abschnitt 4.2].

der im RL häufig genutzte Mechanismus des *Experience Replay* [MKS+15] angewandt werden. Anstelle der Verwendung einzelner Datentupel wird in jedem Trainingsschritt ein sogenannter *Batch* $\{X_0, U_0, R_0, X_1\}$ aus M_B Datentupeln $\{\boldsymbol{x}_k, \boldsymbol{u}_k, r_k, \boldsymbol{x}_{k+1}\}$, die zufällig aus gespeicherten Daten gezogen werden, genutzt. Die Batch-Größe kann dabei prinzipiell für Actor und Critic unterschiedlich sein. Die Verwendung von Experience Replay kann zu einer verringerten Varianz des Trainingsergebnisses führen [MKS+15].

D.3 Wahl der Hyperparameter für das Online-Training

In Tabelle D.1 sind die Hyperparameter für das Online-Training der in Abschnitt 6.1 vorgestellten modellfreien, adaptiven Längsregelung eines realen Fahrzeugs inklusive Vorsteuerung gegeben.

Hyperparameter	Wert
Größe M des Datentupelspeichers	5000
Batch-Größe M_B für Actor und Critic	200
Anzahl h_FIR vergangener Stellgrößen für die Zustandsrekonstruktion	35
Anzahl der Ausgänge des FIR-Filters	1
Anzahl n_h zur Approximation des Sollgeschwindigkeitsverlaufs in (6.8)	2
Amplitude des Anregungssignals auf u_k	$1\,\mathrm{m\,s^{-2}}$
Haltezeit des Anregungssignals	$2\,\mathrm{s}$
Standardabweichung des Anregungsrauschens auf $\boldsymbol{z}_k$	1
Parameter Q_y des Gütefunktionals	1
Parameter R des Gütefunktionals	0,1
Diskontierungsfaktor γ	0,95
Maximale Norm für die Adaption des Actors	0,01
Maximale Norm für die Adaption des Critics	10
Abtastzeit des Systems und Reglers Δt	$0{,}02\,\mathrm{s}$
Updaterate Δl des Trainingsvorgangs	$0{,}6\,\mathrm{s}$

Tabelle D.1: Wahl der Hyperparameter für das Online-Training.

D.4 Ergänzende Messdaten des realen Ball-auf-Platte-Systems

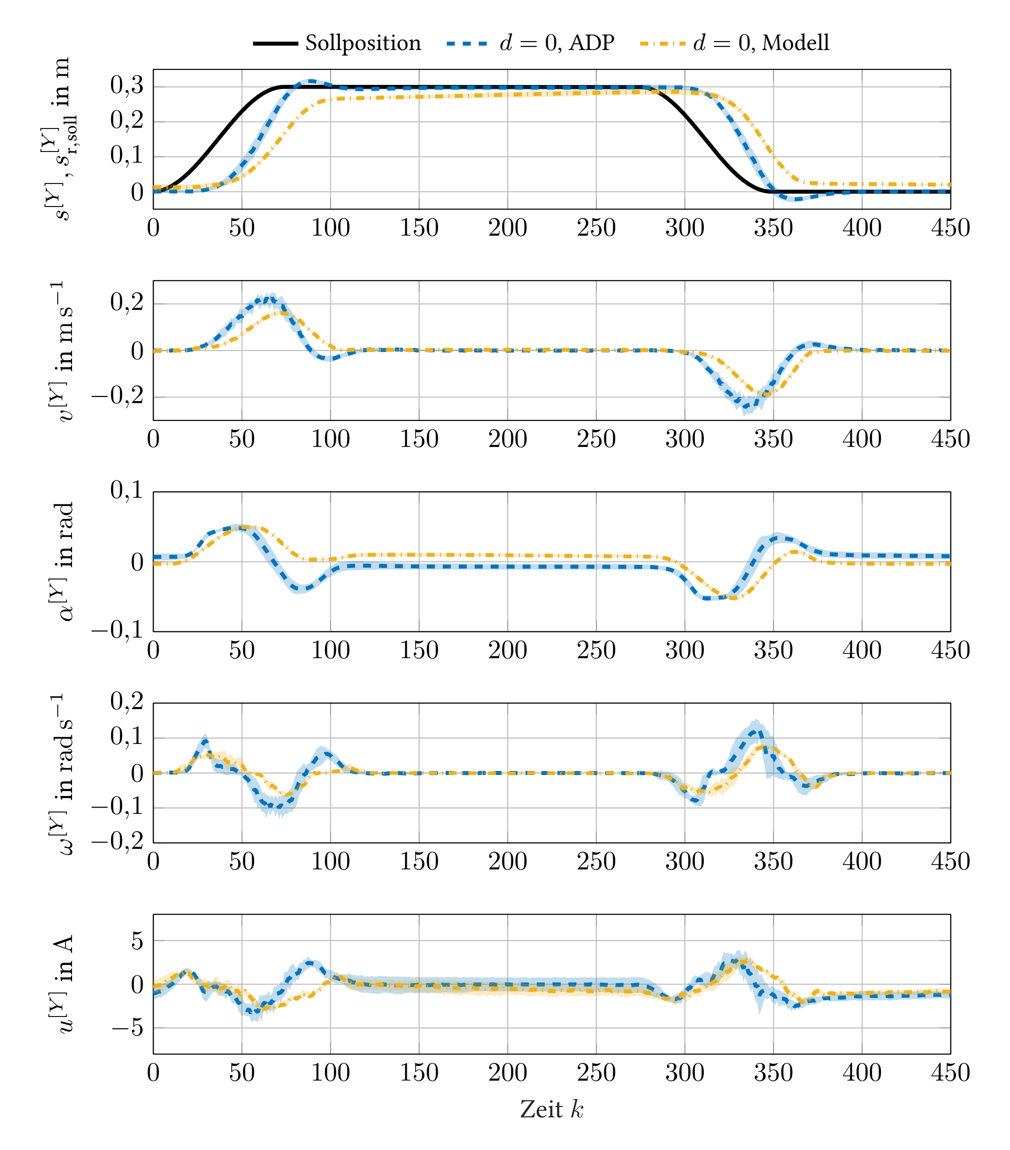


Abbildung D.1: Zustände und Stellgröße bei Vorgabe einer stationären Sollposition ($d = 0$) für den gelernten, ADP-basierten Regler (blau) sowie den modellbasierten Vergleichsregler (gelb).

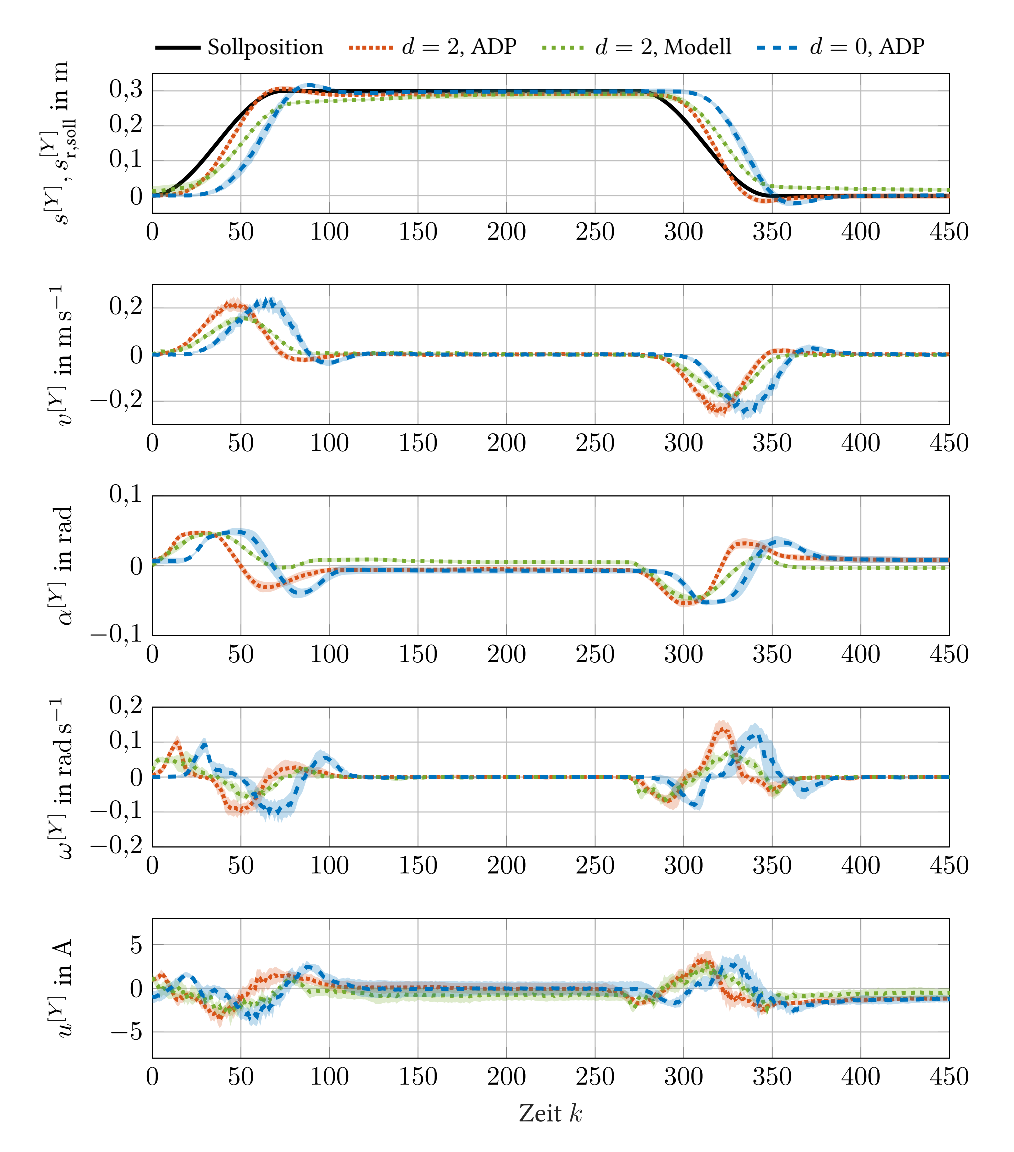


Abbildung D.2: Zustände und Stellgröße bei Vorgabe eines polynomiellen Sollverlaufs ($d = 2$) für den gelernten, ADP-basierten Regler (rot) sowie den modellbasierten Vergleichsregler (grün) im Vergleich zur Vorgabe einer stationären Sollposition ($d = 0$) für den gelernten, ADP-basierten Regler (blau).

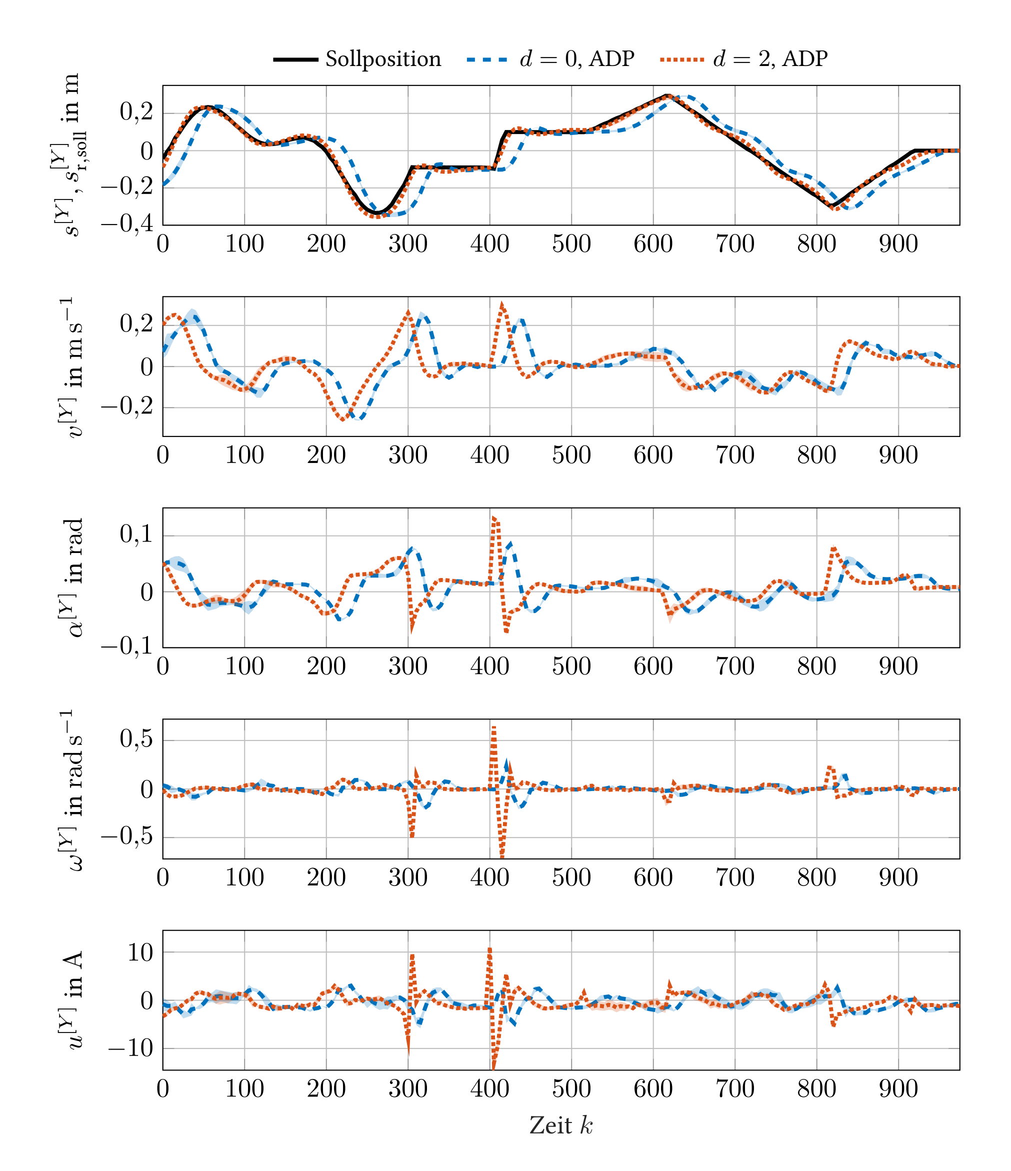


Abbildung D.3: Vergleich der gelernten Solltrajektorienfolgeregler bei Vorgabe eines quadratischen Sollverlaufs ($d = 2$, rot) und einer stationären Sollposition ($d = 0$, blau) für eine Validierungstrajektorie.

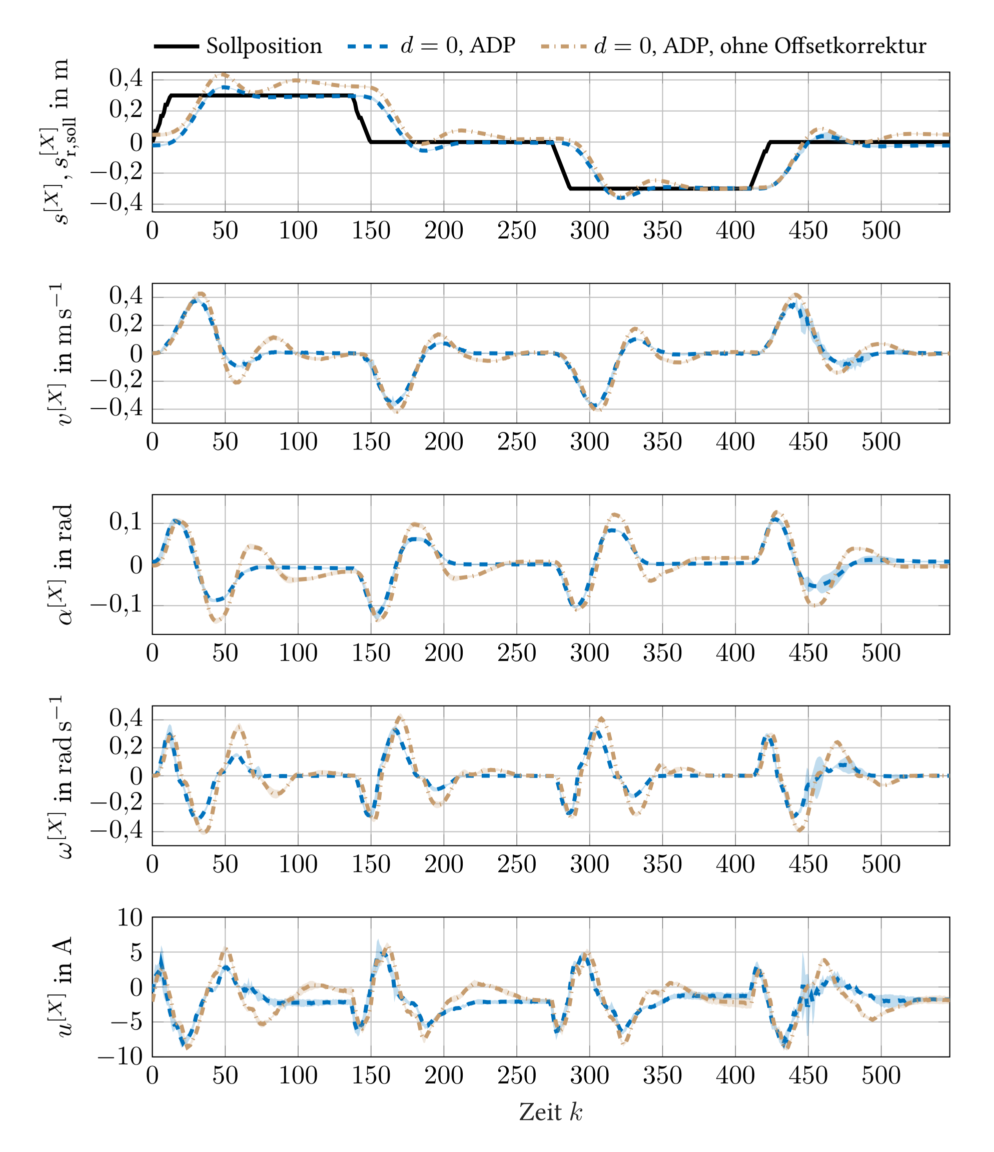


Abbildung D.4: Vergleich eines gelernten Reglers mit (blau) und ohne (braun) lernbare Offsetkorrektur bei Vorgabe einer stationären Sollposition ($d = 0$).

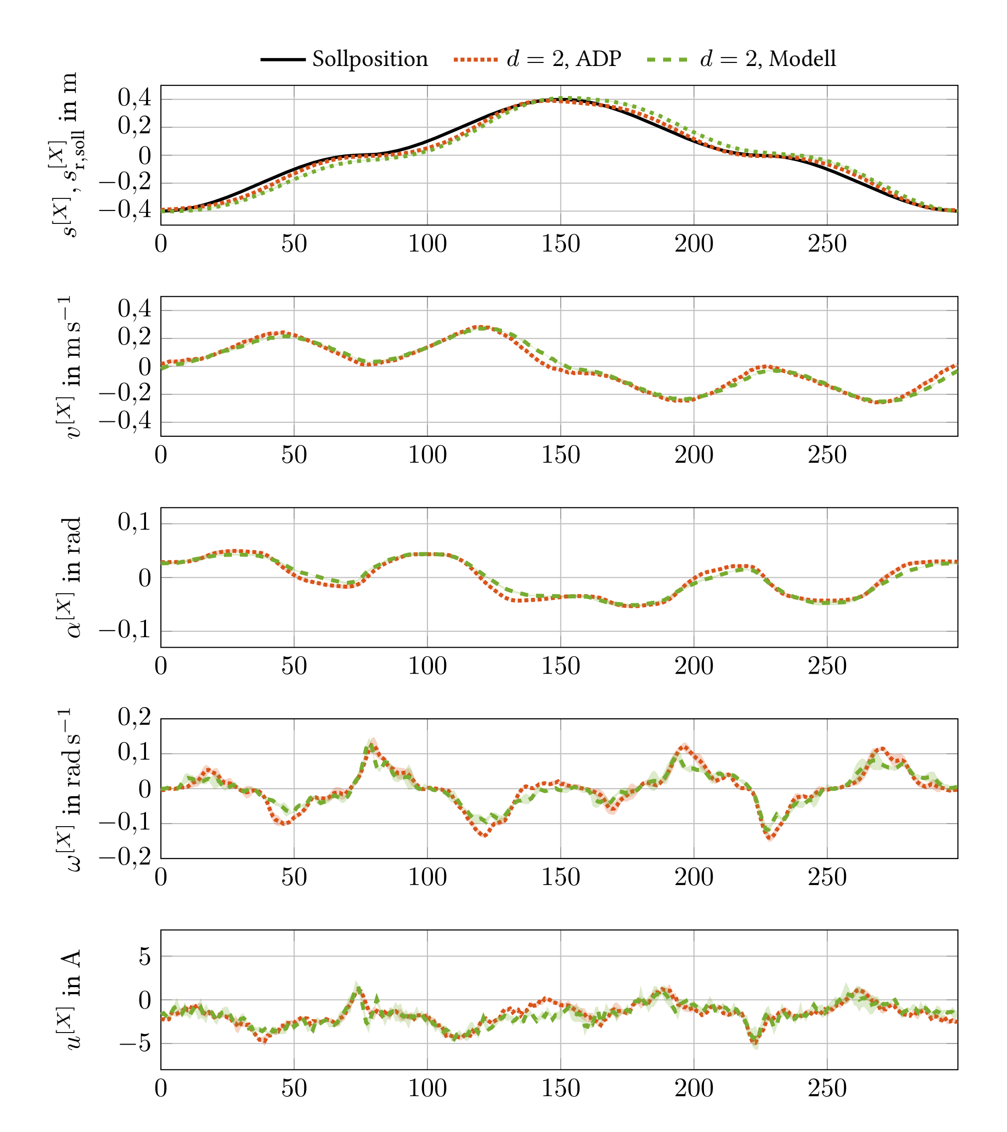


Abbildung D.5: Gleichzeitige Vorgabe eines Sollpositionsverlaufs in beiden Plattendimensionen. Gezeigt sind die resultierenden Zustände und Stellgrößen in X-Richtung für den gelernten Regler und den modellbasierten Vergleichsregler mit jeweils $d = 2$.

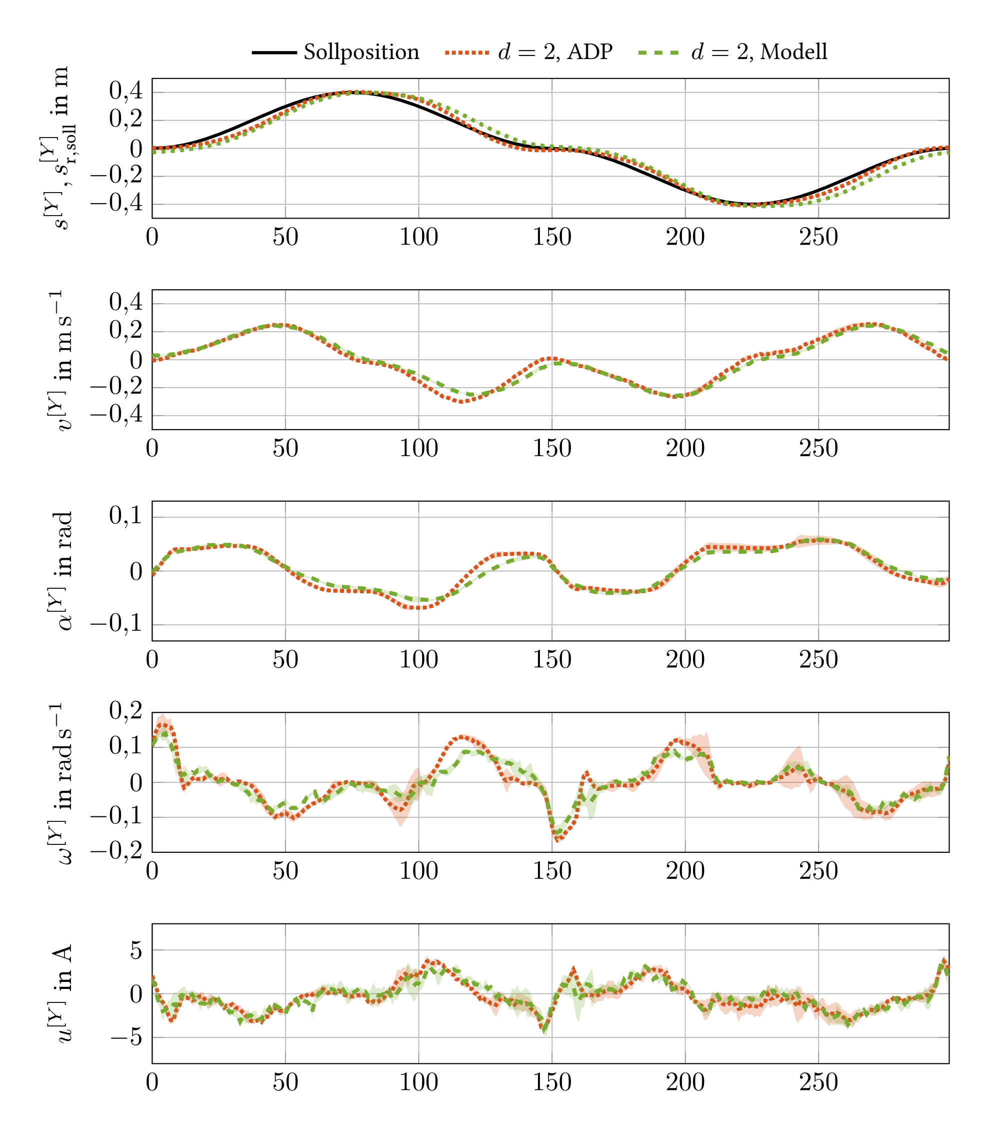


Abbildung D.6: Gleichzeitige Vorgabe eines Sollpositionsverlaufs in beiden Plattendimensionen. Gezeigt sind die resultierenden Zustände und Stellgrößen in Y-Richtung für den gelernten Regler und den modellbasierten Vergleichsregler mit jeweils $d = 2$.

Literaturverzeichnis

Öffentlich zugängliche Quellen

[ÅB66] ÅSTRÖM, K.-J.; BOHLIN, T.: Numerical Identification of Linear Dynamic Systems from Normal Operating Records. In: *IFAC Proceedings Volumes* 2 (1966), Nr. 2, S. 96–111

[ABB+02] AWTAR, S.; BERNARD, C.; BOKLUND, N.; MASTER, A.; UEDA, D.; CRAIG, K.: Mechatronic Design of a Ball-on-Plate Balancing System. In: *Mechatronics* 12 (2002), Nr. 2, S. 217–228

[ABC+20] ANDRYCHOWICZ, M.; BAKER, B.; CHOCIEJ, M.; JÓZEFOWICZ, R.; MCGREW, B.; PACHOCKI, J.; PETRON, A.; PLAPPERT, M.; POWELL, G.; RAY, A.; SCHNEIDER, J.; SIDOR, S.; TOBIN, J.; WELINDER, P.; WENG, L.; ZAREMBA, W.: Learning Dexterous In-Hand Manipulation. In: *The International Journal of Robotics Research* 39 (2020), Nr. 1, S. 3–20

[Ada18] ADAMY, J.: *Nichtlineare Systeme und Regelungen.* 3., aktualisierte Auflage. Berlin: Springer Vieweg, 2018

[AE10] AMANN, H.; ESCHER, J.: *Analysis I.* 3. Auflage. Basel: Birkhäuser, 2010

[AG06] ADETOLA, V.; GUAY, M.: Excitation Signal Design for Parameter Convergence in Adaptive Control of Linearizable Systems. In: *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, S. 447–452

[AG08] ADETOLA, V.; GUAY, M.: Finite-Time Parameter Estimation in Adaptive Control of Nonlinear Systems. In: *IEEE Transactions on Automatic Control* 53 (2008), Nr. 3, S. 807–811

[AKL05] ABU-KHALAF, M.; LEWIS, F. L.: Nearly Optimal Control Laws for Nonlinear Systems with Saturating Actuators Using a Neural Network HJB Approach. In: *Automatica* 41 (2005), Nr. 5, S. 779–791

[AL84] ARNOLD, W. F.; LAUB, A. J.: Generalized Eigenproblem Algorithms and Software for Algebraic Riccati Equations. In: *Proceedings of the IEEE* 72 (1984), Nr. 12, S. 1746–1754

[AM89] ANDERSON, B. D.; MOORE, J. B.: *Optimal Control: Linear Quadratic Methods.* Englewood Cliffs: Prentice Hall, 1989 (Prentice Hall information and system sciences series)

[Apo67] Apostol, T. M.: *Calculus.* 2. Auflage. New York: Wiley, 1967

[ATLAK07] Al-Tamimi, A.; Lewis, F. L.; Abu-Khalaf, M.: Model-Free Q-Learning Designs for Linear Discrete-Time Zero-Sum Games with Application to H-Infinity Control. In: *Automatica* 43 (2007), Nr. 3, S. 473–481

[ATLAK08] Al-Tamimi, A.; Lewis, F. L.; Abu-Khalaf, M.: Discrete-Time Nonlinear HJB Solution Using Approximate Dynamic Programming: Convergence Proof. In: *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics* 38 (2008), Nr. 4, S. 943–949

[ÅW95] Åström, K. J.; Wittenmark, B.: *Adaptive Control.* 2. Auflage. Reading, Mass.: Addison-Wesley, 1995

[Axl15] Axler, S. J.: *Linear Algebra Done Right.* 3. Auflage. Cham, Heidelberg, New York, Dordrecht, London: Springer, 2015

[BA18] Bernhard, S.; Adamy, J.: LQ Optimal Tracking with Unbounded Cost for Unknown Dynamics: An Adaptive Dynamic Programming Approach. In: *European Control Conference.* 2018

[BBdE10] Buşoniu, L.; Babuška, R.; de Schutter, B.; Ernst, D.: *Reinforcement Learning and Dynamic Programming Using Function Approximators.* Boca Raton, London, New York: CRC press, 2010

[BBT+18] Buşoniu, L.; Bruin, T. de; Tolić, D.; Kober, J.; Palunko, I.: Reinforcement Learning for Control: Performance, Stability, and Deep Approximators. In: *Annual Reviews in Control* 46 (2018), S. 8–28

[Bel57a] Bellman, R.: *Dynamic Programming.* 1. Auflage. Princeton: Princeton University Press, 1957

[Bel57b] Bellman, R.: A Markovian Decision Process. In: *Journal of Mathematics and Mechanics* 6 (1957), Nr. 5, S. 679–684

[Ben89] Bentham, J.: *An Introduction to the Principles of Morals and Legislation.* London: T. Payne and Son, 1789

[Ber95] Bertsekas, D. P.: *Dynamic Programming and Optimal Control.* Belmont, Mass.: Athena Scientific, 1995

[Ber17] Bertsekas, D. P.: Value and Policy Iterations in Optimal Control and Adaptive Dynamic Programming. In: *IEEE Transactions on Neural Networks and Learning Systems* 28 (2017), Nr. 3, S. 500–509

[Ber20a] Bernhard, S.: *Optimale Folgeregelung über unendliche Horizonte und optimale Output Regulation für quadratische, über- und unteraktuierte Systeme.* Darmstadt, Technische Universität Darmstadt, Dissertation, 2020

[Ber20b] Bertsekas, D.: Multiagent Value Iteration Algorithms in Dynamic Programming and Reinforcement Learning. In: *Results in Control and Optimization* 1 (2020), S. 1–10

[BFGB12] Braescu, F. C.; Ferariu, L.; Gilca, R.; Bordianu, V.: Ball on Plate Balancing System for Multi-Discipline Educational Purposes. In: *16th International Conference on System Theory, Control and Computing (ICSTCC)*, 2012, S. 1–6

[BHWM12] Burg, K.; Haf, H.; Wille, F.; Meister, A.: *Höhere Mathematik für Ingenieure: Band II: Lineare Algebra.* 7. Auflage. Wiesbaden: Vieweg+Teubner Verlag, 2012

[Bia17] Bian, T.: *Biologically Inspired Adaptive Optimal Control and Learning.* New York, New York University, Dissertation, 2017

[Bit84] Bitmead, R. R.: Persistence of Excitation Conditions and the Convergence of Adaptive Schemes. In: *IEEE Transactions on Information Theory* 30 (1984), Nr. 2, S. 183–191

[BJ16a] Bian, T.; Jiang, Z.-P.: Value Iteration, Adaptive Dynamic Programming, and Optimal Control of Nonlinear Systems. In: *55th IEEE Conference on Decision and Control (CDC)*, 2016, S. 3375–3380

[BJ16b] Bian, T.; Jiang, Z.-P.: Value Iteration and Adaptive Dynamic Programming for Data-Driven Adaptive Optimal Control Design. In: *Automatica* 71 (2016), S. 348–360

[BJD10] Bhasin, S.; Johnson, M.; Dixon, W. E.: A Model-Free Robust Policy Iteration Algorithm for Optimal Control of Nonlinear Systems. In: *49th IEEE Conference on Decision and Control (CDC)*, 2010, S. 3060–3065

[BK18] Büchel, M.; Knoll, A.: Deep Reinforcement Learning for Predictive Longitudinal Control of Automated Vehicles. In: *21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, S. 2391–2397

[BKJ+13] Bhasin, S.; Kamalapurkar, R.; Johnson, M.; Vamvoudakis, K. G.; Lewis, F. L.; Dixon, W. E.: A Novel Actor-Critic-Identifier Architecture for Approximate Optimal Control of Uncertain Nonlinear Systems. In: *Automatica* 49 (2013), Nr. 1, S. 82–92

[BO99] Başar, T.; Olsder, G. J.: *Dynamic Noncooperative Game Theory.* 2. Auflage. Society for Industrial and Applied Mathematics, 1999

[Bro92] Brooks, R. A.: Artificial Life and Real Robots. In: *Proceedings of the First European Conference on Artificial Life*, MIT Press, 1992, S. 3–10

[BS83] Boyd, S.; Sastry, S.: On Parameter Convergence in Adaptive Control. In: *Systems & Control Letters* 3 (1983), Nr. 6, S. 311–319

[BS86] Boyd, S.; Sastry, S. S.: Necessary and Sufficient Conditions for Parameter Convergence in Adaptive Control. In: *Automatica* 22 (1986), Nr. 6, S. 629–639

[BSA83] Barto, A. G.; Sutton, R. S.; Anderson, C. W.: Neuronlike Adaptive Elements that Can Solve Difficult Learning Control Problems. In: *IEEE Transactions on Systems, Man, and Cybernetics* 13 (1983), Nr. 5, S. 834–846

[BSMM08] Bronstein, I. N.; Semendjaev, K. A.; Musiol, G.; Mühlig, H.: *Taschenbuch der Mathematik*. 7., vollständig überarbeitete und ergänzte Auflage. Frankfurt am Main: Verlag Harri Deutsch, 2008

[BSMM13] Bronstein, I. N.; Semendjaev, K. A.; Musiol, G.; Mühlig, H.: *Taschenbuch der Mathematik*. 9. Auflage. Haan-Gruiten: Verlag Europa-Lehrmittel, 2013

[BSW97] Beard, R. W.; Saridis, G. N.; Wen, J. T.: Galerkin Approximations of the Generalized Hamilton-Jacobi-Bellman Equation. In: *Automatica* 33 (1997), Nr. 12, S. 2159–2177

[BT96] Bertsekas, D. P.; Tsitsiklis, J. M.: *Neuro-Dynamic Programming*. Belmont, Mass.: Athena Scientific, 1996

[BU16] Bohn, C.; Unbehauen, H.: *Identifikation dynamischer Systeme*. Wiesbaden: Springer Fachmedien, 2016

[BYB94] Bradtke, S. J.; Ydstie, B. E.; Barto, A. G.: Adaptive Linear Quadratic Control Using Policy Iteration. In: *American Control Conference (ACC)*, 1994, S. 3475–3479

[CDG93] Ciccarella, G.; Dalla Mora, M.; Germani, A.: A Luenberger-like Observer for Nonlinear Systems. In: *International Journal of Control* 57 (1993), Nr. 3, S. 537–556

[CF14] Calanca, A.; Fiorini, P.: Human-adaptive Control of Series Elastic Actuators. In: *Robotica* 32 (2014), Nr. 8, S. 1301–1316

[Che78] Cheney, E. W.: *Introduction to Approximation Theory*. 5. Auflage. New York: McGraw-Hill, 1978 (International Series in Pure and Applied Mathematics)

[CHL19] Collins, J.; Howard, D.; Leitner, J.: Quantifying the Reality Gap in Robotic Manipulation Tasks. In: *International Conference on Robotics and Automation (ICRA)*, 2019, S. 6706–6712

[CJ10] Chowdhary, G.; Johnson, E.: Concurrent Learning for Convergence in Adaptive Control without Persistency of Excitation. In: *49th IEEE Conference on Decision and Control (CDC)*, 2010, S. 3674–3679

[Coh13] Cohn, D. L.: *Measure Theory*. 2. Auflage. Berlin, New York: Springer, 2013

[DAMH19] Dulac-Arnold, G.; Mankowitz, D.; Hester, T.: Challenges of Real-World Reinforcement Learning. In: *36th International Conference on Machine Learning*, 2019

[DCd11] Desjardins, C.; Chaib-draa, B.: Cooperative Adaptive Cruise Control: A Reinforcement Learning Approach. In: *IEEE Transactions on Intelligent Transportation Systems* 12 (2011), Nr. 4, S. 1248–1260

[DCP96] Devasia, S.; Chen, D.; Paden, B.: Nonlinear Inversion-Based Output Tracking. In: *IEEE Transactions on Automatic Control* 41 (1996), Nr. 7, S. 930–942

[Dev02] Devasia, S.: Should Model-Based Inverse Inputs be Used as Feedforward Under Plant Uncertainty? In: *IEEE Transactions on Automatic Control* 47 (2002), Nr. 11, S. 1865–1871

[DHS17] Dušek, F.; Honc, D.; Sharma, K. R.: Modelling of Ball and Plate System Based on First Principle Model and Optimal Control. In: *21st International Conference on Process Control*, 2017, S. 216–221

[DJ09] Dierks, T.; Jagannathan, S.: Optimal Tracking Control of Affine Nonlinear Discrete-Time Systems with Unknown Internal Dynamics. In: *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, 2009, S. 6750–6755

[DJ10] Dierks, T.; Jagannathan, S.: Optimal Control of Affine Nonlinear Continuous-Time Systems. In: *American Control Conference (ACC)*, 2010, S. 1568–1573

[DJ11] Dierks, T.; Jagannathan, S.: Online Optimal Control of Nonlinear Discrete-Time Systems Using Approximate Dynamic Programming. In: *Journal of Control Theory and Applications* 9 (2011), Nr. 3, S. 361–369

[dKTB18] de Bruin, T.; Kober, J.; Tuyls, K.; Babuška, R.: Experience Selection in Deep Reinforcement Learning for Control. In: *Journal of Machine Learning Research* 19 (2018), Nr. 9, S. 1–56

[DLL18] Du, Y.; Liu, C.; Li, Y.: Velocity Control Strategies to Improve Automated Vehicle Driving Comfort. In: *IEEE Intelligent Transportation Systems Magazine* 10 (2018), Nr. 1, S. 8–18

[DLL+19] Duan, J.; Li, S. E.; Liu, Z.; Bujarbaruah, M.; Cheng, B.: Generalized Policy Iteration for Optimal Control in Continuous Time. In: *arXiv preprint 1909.05402* (2019), S. 1–10

[Doy00] Doya, K.: Reinforcement Learning in Continuous Time and Space. In: *Neural Computation* 12 (2000), Nr. 1, S. 219–245

[DS10] Dzjadyk, V. K.; Shevchuk, I. A.: *Theory of Uniform Approximation of Functions by Polynomials*. Berlin, New York: De Gruyter, 2010

[EBHS07] Enns, D.; Bugajski, D.; Hendrick, R.; Stein, G.: Dynamic Inversion: An Evolving Methodology for Flight Control Design. In: *International Journal of Control* 59 (2007), Nr. 1, S. 71–91

[EP10] Eldar, Y. C.; Pohl, V.: Recovering Signals from Lowpass Data. In: *IEEE Transactions on Signal Processing* 58 (2010), Nr. 5, S. 2636–2646

[FC16] Fu, Y.; Chai, T.: Online Solution of Two-Player Zero-Sum Games for Continuous-Time Nonlinear Systems with Completely Unknown Dynamics. In: *IEEE Transactions on Neural Networks and Learning Systems* 27 (2016), Nr. 12, S. 2577–2587

[FFH17] Flad, M.; Fröhlich, L.; Hohmann, S.: Cooperative Shared Control Driver Assistance Systems Based on Motion Primitives and Differential Games. In: *IEEE Transactions on Human-Machine Systems* 47 (2017), Nr. 5, S. 711–722

[Fla16] Flad, M.: *Kooperative Regelungskonzepte auf Basis der Spieltheorie und deren Anwendung auf Fahrerassistenzsysteme.* Karlsruhe, Karlsruher Institut für Technologie, Dissertation, 2016

[FLMR95] Fliess, M.; Lévine, J.; Martin, P.; Rouchon, P.: Flatness and Defect of Nonlinear Systems: Introductory Theory and Examples. In: *International Journal of Control* 61 (1995), Nr. 6, S. 1327–1361

[Föl16] Föllinger, O.: *Regelungstechnik: Einführung in die Methoden und ihre Anwendung.* 12. Auflage. Berlin, Offenbach: VDE-Verlag, 2016

[FOSH14] Flad, M.; Otten, J.; Schwab, S.; Hohmann, S.: Steering Driver Assistance System: A Systematic Cooperative Shared Control Design Approach. In: *IEEE International Conference on Systems, Man and Cybernetics*, 2014, S. 3585–3592

[FWS+18] Feinberg, V.; Wan, A.; Stoica, I.; Jordan, M. I.; Gonzalez, J. E.; Levine, S.: Model-Based Value Expansion for Efficient Model-Free Reinforcement Learning. In: *arXiv preprint 1803.00101* (2018)

[FY16] Fan, Q.-Y.; Yang, G.-H.: Adaptive Actor-Critic Design-Based Integral Sliding-Mode Control for Partially Unknown Nonlinear Systems With Input Disturbances. In: *IEEE Transactions on Neural Networks and Learning Systems* 27 (2016), Nr. 1, S. 165–177

[GBLB12] Grondman, I.; Busoniu, L.; Lopes, G. A. D.; Babuska, R.: A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42 (2012), Nr. 6, S. 1291–1307

[Gee07] Geering, H. P.: *Optimal Control with Engineering Applications.* Berlin, Heidelberg: Springer, 2007

[GGH+18] Gaitsgory, V.; Grüne, L.; Höger, M.; Kellett, C. M.; Weller, S. R.: Stabilization of Strictly Dissipative Discrete Time Systems with Discounted Optimal Control. In: *Automatica* 93 (2018), S. 311–320

[GHLL17] Gu, S.; Holly, E.; Lillicrap, T.; Levine, S.: Deep Reinforcement Learning for Robotic Manipulation with Asynchronous Off-Policy Updates. In: *IEEE International Conference on Robotics and Automation*, 2017, S. 3389–3396

[GJ15] Gao, W.; Jiang, Z.-P.: Linear Optimal Tracking Control: An Adaptive Dynamic Programming Approach. In: *American Control Conference (ACC)*, 2015, S. 4929–4934

[GJ16] Gao, W.; Jiang, Z.-P.: Adaptive Dynamic Programming and Adaptive Optimal Output Regulation of Linear Systems. In: *IEEE Transactions on Automatic Control* 61 (2016), Nr. 12, S. 4164–4169

[Gli11] Glimcher, P. W.: Understanding Dopamine and Reinforcement Learning: The Dopamine Reward Prediction Error Hypothesis. In: *Proceedings of the National Academy of Sciences of the United States of America* (2011), S. 15647–15654

[GM86] Green, M.; Moore, J. B.: Persistence of Excitation in Linear Systems. In: *Systems & Control Letters* 7 (1986), Nr. 5, S. 351–360

[Gör17] Görges, D.: Relations between Model Predictive Control and Reinforcement Learning. In: *IFAC-PapersOnLine* 50 (2017), Nr. 1, S. 4920–4928

[Gör19] Görges, D.: Distributed Adaptive Linear Quadratic Control using Distributed Reinforcement Learning. In: *IFAC-PapersOnLine* 52 (2019), Nr. 11, S. 218–223

[Gos09] Gosavi, A.: Reinforcement Learning: A Tutorial Survey and Recent Advances. In: *INFORMS Journal on Computing* 21 (2009), Nr. 2, S. 178–192

[GSK19] Gavrishchaka, V.; Senyukova, O.; Koepke, M.: Synergy of Physics-Based Reasoning and Machine Learning in Biomedical Applications: Towards Unlimited Deep Learning with Limited Data. In: *Advances in Physics: X* 4 (2019), Nr. 1, S. 204–256

[HDB94] Houk, J. C.; Davis, J. L.; Beiser, D. G.: A Model of How the Basal Ganglia Generate and Use Neural Signals that Predict Reinforcement. In: *Models of Information Processing in the Basal Ganglia*. 1994, S. 249–270

[Hey16] Heydari, A.: Analyzing Policy Iteration in Optimal Control. In: *American Control Conference (ACC)*, 2016, S. 5728–5733

[Hir79] Hirschorn, R. M.: Invertibility of Nonlinear Control Systems. In: *SIAM Journal on Control and Optimization* 17 (1979), Nr. 2, S. 289–297

[HJK20] Hyatt, P.; Johnson, C. C.; Killpack, M. D.: Model Reference Predictive Adaptive Control for Large-Scale Soft Robots. In: *Frontiers in Robotics and AI* 7 (2020), 132

[HL14] Huang, Y.; Liu, D.: Neural-Network-Based Optimal Tracking Control Scheme for a Class of Unknown Discrete-Time Nonlinear Systems Using Iterative ADP Algorithm. In: *Neurocomputing* 125 (2014), S. 46–56

[HLM03] Hespanha, J. P.; Liberzon, D.; Morse, A.: Overcoming the Limitations of Adaptive Control by Means of Logic-Based Switching. In: *Systems & Control Letters* 49 (2003), Nr. 1, S. 49–65

[HMT+21] Hamaya, M.; Matsubara, T.; Teramae, T.; Noda, T.; Morimoto, J.: Design of Physical User-Robot Interactions for Model Identification of Soft Actuators on Exoskeleton Robots. In: *The International Journal of Robotics Research* 40 (2021), Nr. 1, S. 397–410

[HS15] Hausknecht, M.; Stone, P.: Deep Recurrent Q-Learning for Partially Observable MDPs. In: *AAAI Fall Symposium Series*, 2015, S. 1–7

[HSSH17] Hwangbo, J.; Sa, I.; Siegwart, R.; Hutter, M.: Control of a Quadrotor with Reinforcement Learning. In: *IEEE Robotics and Automation Letters* 2 (2017), Nr. 4, S. 2096–2103

[HSW90] Hornik, K.; Stinchcombe, M.; White, H.: Universal Approximation of an Unknown Mapping and its Derivatives Using Multilayer Feedforward Networks. In: *Neural Networks* 3 (1990), Nr. 5, S. 551–560

[HWL21] Ha, M.; Wang, D.; Liu, D.: Generalized Value Iteration for Discounted Optimal Control with Stability Analysis. In: *Systems & Control Letters* 147 (2021), S. 1–7

[HXH+19] Huang, Z.; Xu, X.; He, H.; Tan, J.; Sun, Z.: Parameterized Batch Reinforcement Learning for Longitudinal Control of Autonomous Land Vehicles. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49 (2019), Nr. 4, S. 730–741

[IS96] Ioannou, P. A.; Sun, J.: *Robust Adaptive Control.* Upper Saddle River, New Jersey: Prentice-Hall Inc., 1996

[Isi89] Isidori, A.: *Nonlinear Control Systems.* 2. Auflage. Berlin, Heidelberg: Springer, 1989

[JALG18] Jenkins, B. M.; Annaswamy, A. M.; Lavretsky, E.; Gibson, T. E.: Convergence Properties of Adaptive Systems and the Definition of Exponential Stability. In: *SIAM Journal on Control and Optimization* 56 (2018), Nr. 4, S. 2463–2484

[JJ12] Jiang, Y.; Jiang, Z.-P.: Computational Adaptive Optimal Control for Continuous-Time Linear Systems with Completely Unknown Dynamics. In: *Automatica* 48 (2012), Nr. 10, S. 2699–2704

[JJ14a] Jiang, Y.; Jiang, Z.-P.: Adaptive Dynamic Programming as a Theory of Sensorimotor Control. In: *Biological Cybernetics* 108 (2014), Nr. 4, S. 459–473

[JJ14b] Jiang, Y.; Jiang, Z.-P.: Global Adaptive Dynamic Programming for Continuous-Time Nonlinear Polynomial Systems. In: *IFAC Proceedings Volumes* 47 (2014), Nr. 3, S. 9756–9761

[JJ14c] Jiang, Y.; Jiang, Z.-P.: Robust Adaptive Dynamic Programming and Feedback Stabilization of Nonlinear Systems. In: *IEEE Transactions on Neural Networks and Learning Systems* 25 (2014), Nr. 5, S. 882–893

[JKBD15] Johnson, M.; Kamalapurkar, R.; Bhasin, S.; Dixon, W. E.: Approximate N-Player Nonzero-Sum Game Solution for an Uncertain Continuous Nonlinear System. In: *IEEE Transactions on Neural Networks and Learning Systems* 26 (2015), Nr. 8, S. 1645–1658

[JZLH19] Jiang, H.; Zhang, H.; Luo, Y.; Han, J.: Neural-Network-Based Robust Control Schemes for Nonlinear Multiplayer Systems with Uncertainties via Adaptive Dynamic Programming. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49 (2019), Nr. 3, S. 579–588

[KAV+14] Katzourakis, D. I.; Abbink, D. A.; Velenis, E.; Holweg, E.; Happee, R.: Drivers Arms Time-Variant Neuromuscular Admittance During Real Car Test-Track Driving. In: *IEEE Transactions on Instrumentation and Measurement* 63 (2014), Nr. 1, S. 221–230

[KB18] Knabner, P.; Barth, W.: *Lineare Algebra: Grundlagen und Anwendungen.* 2. Auflage. Berlin, Heidelberg: Springer, 2018

[KBJ+21] Kuutti, S.; Bowden, R.; Jin, Y.; Barber, P.; Fallah, S.: A Survey of Deep Learning Applications to Autonomous Vehicle Control. In: *IEEE Transactions on Intelligent Transportation Systems* 22 (2021), Nr. 2, S. 712–733

[KBP13] Kober, J.; Bagnell, J. A.; Peters, J.: Reinforcement Learning in Robotics: A Survey. In: *The International Journal of Robotics Research* 32 (2013), Nr. 11, S. 1238–1274

[KCS03] Knuplez, A.; Chowdhury, A.; Svecko, R.: Modeling and Control Design for the Ball and Plate System. In: *IEEE International Conference on Industrial Technology*, 2003, S. 1064–1067

[KDBD15] Kamalapurkar, R.; Dinh, H.; Bhasin, S.; Dixon, W. E.: Approximate Optimal Trajectory Tracking for Continuous-Time Nonlinear Systems. In: *Automatica* 51 (2015), S. 40–48

[Kha02] Khalil, H. K.: *Nonlinear Systems.* 3. Auflage. Upper Saddle River: Prentice Hall, 2002

[KHL+12] Khan, S. G.; Herrmann, G.; Lewis, F. L.; Pipe, T.; Melhuish, C.: Reinforcement Learning and Optimal Adaptive Control: An Overview and Implementation Examples. In: *Annual Reviews in Control* 36 (2012), Nr. 1, S. 42–59

[KIP+18] Kalashnikov, D.; Irpan, A.; Pastor, P.; Ibarz, J.; Herzog, A.; Jang, E.; Quillen, D.; Holly, E.; Kalakrishnan, M.; Vanhoucke, V.; Levine, S.: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. In: *Proceedings of Machine Learning Research* Bd. 87. 2018, S. 651–673

[KKD14] Kamalapurkar, R.; Klotz, J. R.; Dixon, W. E.: Concurrent Learning-Based Approximate Feedback-Nash Equilibrium Solution of N-Player Nonzero-Sum Differential Games. In: *IEEE/CAA Journal of Automatica Sinica* 1 (2014), Nr. 3, S. 239–247

[KL15] Kiumarsi, B.; Lewis, F. L.: Actor-Critic-Based Optimal Tracking for Partially Unknown Nonlinear Discrete-Time Systems. In: *IEEE Transactions on Neural Networks and Learning Systems* 26 (2015), Nr. 1, S. 140–151

[KLJ17] Kiumarsi, B.; Lewis, F. L.; Jiang, Z.-P.: H∞ Control of Linear Discrete-Time Systems: Off-Policy Reinforcement Learning. In: *Automatica* 78 (2017), S. 144–152

[KLL15] Kiumarsi, B.; Lewis, F. L.; Levine, D. S.: Optimal Control of Nonlinear Discrete Time-Varying Systems Using a New Neural Network Approximation Structure. In: *Neurocomputing* 156 (2015), S. 157–165

[KLM96] Kaelbling, L. P.; Littman, M. L.; Moore, A. W.: Reinforcement Learning: A Survey. In: *Journal of Artificial Intelligence Research* 4 (1996), Nr. 1, S. 237–285

[KLM+14] Kiumarsi, B.; Lewis, F. L.; Modares, H.; Karimpour, A.; Naghibi-Sistani, M.-B.: Reinforcement Q-Learning for Optimal Tracking Control of Linear Discrete-Time Systems with Unknown Dynamics. In: *Automatica* 50 (2014), Nr. 4, S. 1167–1175

[KLNSK15] Kiumarsi, B.; Lewis, F. L.; Naghibi-Sistani, M.; Karimpour, A.: Optimal Tracking Control of Unknown Discrete-Time Linear Systems Using Input-Output Measured Data. In: *IEEE Transactions on Cybernetics* 45 (2015), Nr. 12, S. 2770–2779

[KT03] Konda, V. R.; Tsitsiklis, J. N.: On Actor-Critic Algorithms. In: *SIAM Journal on Control and Optimization* 42 (2003), Nr. 4, S. 1143–1166

[KT04] Kulikov, G. G. (Hrsg.); Thompson, H. A. (Hrsg.): *Dynamic Modelling of Gas Turbines: Identification, Simulation, Condition Monitoring and Optimal Control*. London: Springer, 2004 (Advances in Industrial Control)

[Kuč72] Kučera, V.: The Discrete Riccati Equation of Optimal Control. In: *Kybernetika* 8 (1972), Nr. 5, S. 430–447

[Kuč73] Kučera, V.: A Review of the Matrix Riccati Equation. In: *Kybernetika* 9 (1973), Nr. 5, S. 42–61

[KWD13] Kamalapurkar, R.; Walters, P.; Dixon, W.: Concurrent Learning-Based Approximate Optimal Regulation. In: *52nd IEEE Conference on Decision and Control (CDC)*, 2013, S. 6256–6261

[KWD16] Kamalapurkar, R.; Walters, P.; Dixon, W. E.: Model-Based Reinforcement Learning for Approximate Optimal Regulation. In: *Automatica* 64 (2016), S. 94–104

[Lan97] Landelius, T.: *Reinforcement Learning and Distributed Local Model Synthesis.* Linköping, Linköping University, Dissertation, 1997

[LCL+19] Li, J.; Chai, T.; Lewis, F. L.; Ding, Z.; Jiang, Y.: Off-Policy Interleaved Q-Learning: Optimal Control for Affine Nonlinear Discrete-Time Systems. In: *IEEE Transactions on Neural Networks and Learning Systems* 30 (2019), Nr. 5, S. 1308–1320

[Lév09] Lévine, J.: *Analysis and Control of Nonlinear Systems. A Flatness-Based Approach.* Berlin, Heidelberg: Springer, 2009

[LG19] Li, G.; Görges, D.: Ecological Adaptive Cruise Control and Energy Management Strategy for Hybrid Electric Vehicles Based on Heuristic Dynamic Programming. In: *IEEE Transactions on Intelligent Transportation Systems* 20 (2019), Nr. 9, S. 3526–3535

[LG20] Li, G.; Görges, D.: Ecological Adaptive Cruise Control for Vehicles With Step-Gear Transmission Based on Reinforcement Learning. In: *IEEE Transactions on Intelligent Transportation Systems* 21 (2020), Nr. 11, S. 4895–4905

[LGM20] Li, G.; Görges, D.; Mu, C.: Integrated Adaptive Dynamic Programming for Data-Driven Optimal Controller Design. In: *Neurocomputing* 403 (2020), S. 143–152

[LHP+16] Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D.: Continuous Control with Deep Reinforcement Learning. In: *4th International Conference on Learning Representations*, 2016, S. 1–14

[Lib12] Liberzon, D.: *Calculus of Variations and Optimal Control Theory.* Princeton: Princeton University Press, 2012

[LK98] Lin, J.-S.; Kanellakopoulos, I.: Nonlinearities Enhance Parameter Convergence in Output-Feedback Systems. In: *IEEE Transactions on Automatic Control* 43 (1998), Nr. 2, S. 204–222

[LK99] Lin, J.-S.; Kanellakopoulos, I.: Nonlinearities Enhance Parameter Convergence in Strict Feedback Systems. In: *IEEE Transactions on Automatic Control* 44 (1999), Nr. 1, S. 89–94

[LL04] Lee, J. M.; Lee, J. H.: Approximate Dynamic Programming Strategies and Their Applicability for Process Control: A Review and Future Directions. In: *International Journal of Control Automation and Systems* 2 (2004), S. 263–278

[LLHW16] Luo, B.; Liu, D.; Huang, T.; Wang, D.: Model-Free Optimal Tracking Control via Critic-Only Q-Learning. In: *IEEE Transactions on Neural Networks and Learning Systems* 27 (2016), Nr. 10, S. 2134–2144

[LLSX14] Lu, R.; Li, Z.; Su, C.-Y.; Xue, A.: Development and Learning Control of a Human Limb with a Rehabilitation Exoskeleton. In: *IEEE Transactions on Industrial Electronics* 61 (2014), Nr. 7, S. 3776–3785

[LLW14] Liu, D.; Li, H.; Wang, D.: Online Synchronous Approximate Optimal Learning Algorithm for Multi-Player Non-Zero-Sum Games with Unknown Dynamics. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 44 (2014), Nr. 8, S. 1015–1027

[LLW+17] Luo, B.; Liu, D.; Wu, H.-N.; Wang, D.; Lewis, F. L.: Policy Gradient Adaptive Dynamic Programming for Data-Based Optimal Control. In: *IEEE Transactions on Cybernetics* 47 (2017), Nr. 10, S. 3341–3354

[LNY+15] Lv, Y.; Na, J.; Yang, Q.; Wu, X.; Guo, Y.: Online Adaptive Optimal Control for Continuous-Time Nonlinear Systems with Completely Unknown Dynamics. In: *International Journal of Control* 89 (2015), Nr. 1, S. 99–112

[LP03] Lagoudakis, M. G.; Parr, R.: Least-Squares Policy Iteration. In: *Journal of Machine Learning Research* 4 (2003), S. 1107–1149

[LPC12] Lee, J. Y.; Park, J. B.; Choi, Y. H.: Integral Q-Learning and Explorized Policy Iteration for Adaptive Optimal Control of Continuous-Time Linear Systems. In: *Automatica* 48 (2012), Nr. 11, S. 2850–2859

[LPC15] Lee, J. Y.; Park, J. B.; Choi, Y. H.: Integral Reinforcement Learning for Continuous-Time Input-Affine Nonlinear Systems with Simultaneous Invariant Explorations. In: *IEEE Transactions on Neural Networks and Learning Systems* 26 (2015), Nr. 5, S. 916–932

[LS17] Lee, J. Y.; Sutton, R. S.: Integral Policy Iterations for Reinforcement Learning Problems in Continuous Time and Space. In: *arXiv preprint 1705.03520* (2017)

[Lun20a] Lunze, J.: *Regelungstechnik 1.* 12., überarbeitete Auflage. Berlin, Heidelberg: Springer Vieweg, 2020

[Lun20b] Lunze, J.: *Regelungstechnik 2.* 10., überarbeitete und aktualisierte Auflage. Berlin, Heidelberg: Springer Vieweg, 2020

[LV09] Lewis, F.; Vrabie, D.: Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control. In: *IEEE Circuits and Systems Magazine* 9 (2009), Nr. 3, S. 32–50

[LVS12] Lewis, F. L.; Vrabie, D. L.; Syrmos, V. L.: *Optimal Control.* 3. Auflage. Hoboken: Wiley, 2012

[LVV12] Lewis, F. L.; Vrabie, D.; Vamvoudakis, K. G.: Reinforcement Learning and Feedback Control: Using Natural Decision Methods to Design Optimal Adaptive Controllers. In: *IEEE Control Systems Magazine* 32 (2012), Nr. 6, S. 76–105

[LW14] Liu, D.; Wei, Q.: Policy Iteration Adaptive Dynamic Programming Algorithm for Discrete-Time Nonlinear Systems. In: *IEEE Transactions on Neural Networks and Learning Systems* 25 (2014), Nr. 3, S. 621–634

[LWW+17] Liu, D.; Wei, Q.; Wang, D.; Yang, X.; Li, H.: *Adaptive Dynamic Programming with Applications in Optimal Control.* Cham: Springer International Publishing, 2017 (Advances in Industrial Control)

[LYD17] Li, J.; Yuan, D.; Ding, Z.: Optimal Tracking Control for Discrete-Time Systems by Model-Free Off-Policy Q-Learning Approach. In: *11th Asian Control Conference*, 2017, S. 7–12

[LYWW15] Liu, D.; Yang, X.; Wang, D.; Wei, Q.: Reinforcement-Learning-Based Robust Controller Design for Continuous-Time Uncertain Nonlinear Systems Subject to Input Constraints. In: *IEEE Transactions on Cybernetics* 45 (2015), Nr. 7, S. 1372–1385

[Mar84] Mareels, I.: Sufficiency of Excitation. In: *Systems & Control Letters* 5 (1984), Nr. 3, S. 159–163

[MB90] Mathelin, M. de; Bodson, M.: Frequency Domain Conditions for Parameter Convergence in Multivariable Recursive Identification. In: *Automatica* 26 (1990), Nr. 4, S. 757–767

[MBK19] MacCluer, B. D.; Bourdon, P.; Kriete, T.: *Differential Equations: Techniques, Theory, and Applications.* Providence: American Mathematical Society, 2019

[MBTL12] Mannava, A.; Balakrishnan, S. N.; Tang, L.; Landers, R. G.: Optimal Tracking Control of Motion Systems. In: *IEEE Transactions on Control Systems Technology* 20 (2012), Nr. 6, S. 1548–1558

[MCLS02] Murray, J. J.; Cox, C. J.; Lendaris, G. G.; Saeks, R.: Adaptive Dynamic Programming. In: *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews* 32 (2002), Nr. 2, S. 140–153

[Mey00] Meyer, C. D.: *Matrix Analysis and Applied Linear Algebra.* Philadelphia: SIAM, 2000

[MKS+13] Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M.: Playing Atari with Deep Reinforcement Learning. In: *arXiv preprint 1312.5602* (2013), S. 1–9

[MKS+15] Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; Hassabis, D.: Human-Level Control Through Deep Reinforcement Learning. In: *Nature* 518 (2015), S. 529–533

[ML13] Modares, H.; Lewis, F. L.: Online Solution to the Linear Quadratic Tracking Problem of Continuous-Time Systems Using Reinforcement Learning. In: *52nd IEEE Conference on Decision and Control*, 2013, S. 3851–3856

[ML14a] Modares, H.; Lewis, F. L.: Linear Quadratic Tracking Control of Partially-Unknown Continuous-Time Systems Using Reinforcement Learning. In: *IEEE Transactions on Automatic Control* 59 (2014), Nr. 11, S. 3051–3056

[ML14b] Modares, H.; Lewis, F. L.: Optimal Tracking Control of Nonlinear Partially-Unknown Constrained-Input Systems Using Integral Reinforcement Learning. In: *Automatica* 50 (2014), Nr. 7, S. 1780–1792

[MM70] Mendel, J. M.; McLaren, R. W.: Reinforcement-Learning Control and Pattern Recognition Systems. In: Mendel, J. M. (Hrsg.); Fu, K. S. (Hrsg.): *Adaptive, Learning and Pattern Recognition Systems.* New York: Academic Press, 1970, S. 287–318

[MMP+10] Michalos, G.; Makris, S.; Papakostas, N.; Mourtzis, D.; Chryssolouris, G.: Automotive Assembly Technologies Review: Challenges and Outlook for a Flexible and Adaptive Approach. In: *CIRP Journal of Manufacturing Science and Technology* 2 (2010), Nr. 2, S. 81–91

[MN77] Morgan, A. P.; Narendra, K. S.: On the Uniform Asymptotic Stability of Certain Linear Nonautonomous Differential Equations. In: *SIAM Journal on Control and Optimization* 15 (1977), Nr. 1, S. 5–24

[Mor78] Moré, J. J.: The Levenberg-Marquardt Algorithm: Implementation and Theory. In: Watson, G. A. (Hrsg.): *Numerical Analysis.* Berlin, Heidelberg: Springer, 1978, S. 105–116

[MRLP16] Modares, H.; Ranatunga, I.; Lewis, F. L.; Popa, D. O.: Optimized Assistive Human-Robot Interaction Using Reinforcement Learning. In: *IEEE Transactions on Cybernetics* 46 (2016), Nr. 3, S. 655–667

[MSV08] Moarref, M.; Saadat, M.; Vossoughi, G.: Mechatronic Design and Position Control of a Novel Ball and Plate System. In: *16th Mediterranean Conference on Control and Automation*, 2008, S. 1071–1076

[MSWS17] Mu, C.; Sun, C.; Wang, D.; Song, A.: Adaptive Tracking Control for a Class of Continuous-Time Uncertain Nonlinear Systems Using the Approximate Solution of HJB Equation. In: *Neurocomputing* 260 (2017), S. 432–442

[Mun06] Munos, R.: Policy Gradient in Continuous Time. In: *Journal of Machine Learning Research* 7 (2006), Nr. 27, S. 771–791

[NA87] Narendra, K. S.; Annaswamy, A. M.: Persistent Excitation in Adaptive Systems. In: *International Journal of Control* 45 (1987), Nr. 1, S. 127–160

[NA05] Narendra, K. S.; Annaswamy, A. M.: *Stable Adaptive Systems*. 2. Auflage. Mineola, New York: Dover Publications, 2005

[NC15] Na, X.; Cole, D. J.: Game-Theoretic Modeling of the Steering Interaction Between a Human Driver and a Vehicle Collision Avoidance Controller. In: *IEEE Transactions on Human-Machine Systems* 45 (2015), Nr. 1, S. 25–38

[NCH08] Ng, L.; Clark, C. M.; Huissoon, J. P.: Reinforcement Learning of Adaptive Longitudinal Vehicle Control for Dynamic Collaborative Driving. In: *IEEE Intelligent Vehicles Symposium*, 2008, S. 907–912

[Ngu18] Nguyen, N. T.: *Model-Reference Adaptive Control*. Cham: Springer International Publishing, 2018

[Niv09] Niv, Y.: Reinforcement Learning in the Brain. In: *Journal of Mathematical Psychology* 53 (2009), Nr. 3, S. 139–154

[NKJS04] Ng, A. Y.; Kim, H. J.; Jordan, M. I.; Sastry, S.: Autonomous Helicopter Flight via Reinforcement Learning. In: Thrun, S. (Hrsg.); Saul, L. K. (Hrsg.); Schölkopf, B. (Hrsg.): *Advances in Neural Information Processing Systems*. Cambridge: MIT Press, 2004, S. 799–806

[NLW+19] Na, J.; Li, G.; Wang, B.; Herrmann, G.; Zhan, S.: Robust Optimal Control of Wave Energy Converters Based on Adaptive Dynamic Programming. In: *IEEE Transactions on Sustainable Energy* 10 (2019), Nr. 2, S. 961–970

[NMH+15] Na, J.; Mahyuddin, M. N.; Herrmann, G.; Ren, X.; Barber, P.: Robust Adaptive Finite-Time Parameter Estimation and Control for Robotic Systems. In: *International Journal of Robust and Nonlinear Control* 25 (2015), Nr. 16, S. 3045–3071

[NP96] Nevistic, V.; Primbs, J. A.: *Constrained Nonlinear Optimal Control: A Converse HJB Approach*. Pasadena, California Institute of Technology, Technischer Bericht, 1996

[NW06] Nocedal, J.; Wright, S. J.: *Numerical Optimization*. 2. Auflage. New York: Springer Science and Business Media LLC, 2006

[Pav27] Pavlov, I. P.: *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex*. Übersetzt durch G. V. Anrep. London: Oxford University Press, 1927

[Pla19] PLATTFORM INDUSTRIE 4.0; BUNDESMINISTERIUM FÜR WIRTSCHAFT UND ENERGIE (Hrsg.): *Technologieszenario Künstliche Intelligenz in der Industrie 4.0*. 2019

[PLB15] PAPAGEORGIOU, M.; LEIBOLD, M.; BUSS, M.: *Optimierung: Statische, dynamische, stochastische Verfahren für die Anwendung*. 4. Auflage. Berlin, Heidelberg: Springer, 2015

[PMB13] PASCANU, R.; MIKOLOV, T.; BENGIO, Y.: On the Difficulty of Training Recurrent Neural Networks. In: *Proceedings of the 30th International Conference on Machine Learning*, 2013, S. 1310–1318

[PMK99] PESHKIN, L.; MEULEAU, N.; KAELBLING, L. P.: Learning Policies with External Memory. In: *Proceedings of the 16th International Conference on Machine Learning*, 1999, S. 307–314

[Pow11] POWELL, P. D.: Calculating Determinants of Block Matrices. In: *arXiv preprint 1112.4379* (2011), S. 1–11

[Pra17] PRALY, L.: *Convergence of the Gradient Algorithm for Linear Regression Models in the Continuous and Discrete Time Cases*. Paris, Mines ParisTech, Technischer Bericht, 2017

[PRH19] PUCCETTI, L.; RATHGEBER, C.; HOHMANN, S.: Actor-Critic Reinforcement Learning for Linear Longitudinal Output Control of a Road Vehicle. In: *IEEE Intelligent Transportation Systems Conference*, 2019, S. 2907–2913

[PSA17] PADOAN, A.; SCARCIOTTI, G.; ASTOLFI, A.: A Geometric Characterization of the Persistence of Excitation Condition for the Solutions of Autonomous Systems. In: *IEEE Transactions on Automatic Control* 62 (2017), Nr. 11, S. 5666–5677

[PT12] PIETQUIN, O.; TANGO, F.: A Reinforcement Learning Approach to Optimize the Longitudinal Behavior of a Partial Autonomous Driving Assistance System. In: *Proceedings of the 20th European Conference on Artificial Intelligence*, 2012, S. 987–992

[QZL13] QIN, C.; ZHANG, H.; LUO, Y.: Online Optimal Tracking Control of Continuous-Time Linear Systems with Unknown Dynamics by Using Adaptive Dynamic Programming. In: *International Journal of Control* 87 (2013), Nr. 5, S. 1000–1009

[QZLY19] QU, Q.; ZHANG, H.; LUO, C.; YU, R.: Robust Control Design for Multi-Player Nonlinear Systems with Input Disturbances via Adaptive Dynamic Programming. In: *Neurocomputing* 334 (2019), S. 1–10

[Rec19] RECHT, B.: A Tour of Reinforcement Learning: The View from Continuous Control. In: *Annual Review of Control, Robotics, and Autonomous Systems* 2 (2019), S. 253–279

[RPS07] Riedmiller, M.; Peters, J.; Schaal, S.: Evaluation of Policy Gradient Methods and Variants on the Cart-Pole Benchmark. In: *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, 2007, S. 254–261

[RW72] Rescorla, R. A.; Wagner, A. R.: A Theory of Pavlovian Conditioning: Variations on the Effectiveness of Reinforcement and Non-Reinforcement. In: Black, A. H. (Hrsg.); Prokasy, W. F. (Hrsg.): *Classical Conditioning II: Current Research and Theory.* New York: Appleton-Century-Crofts, 1972, S. 64–99

[SB89] Sastry, S.; Bodson, M.: *Adaptive Control: Stability, Convergence, and Robustness.* Englewood Cliffs: Prentice-Hall, 1989

[SB18] Sutton, R. S.; Barto, A. G.: *Reinforcement Learning: An Introduction.* 2. Auflage. Cambridge, Mass.: MIT Press, 2018

[SBW92] Sutton, R. S.; Barto, A. G.; Williams, R. J.: Reinforcement Learning is Direct Adaptive Optimal Control. In: *IEEE Control Systems Magazine* 12 (1992), Nr. 2, S. 19–22

[SCN+04] Smets, I. Y.; Claes, J. E.; November, E. J.; Bastin, G. P.; van Impe, J. F.: Optimal Adaptive Control of (Bio)chemical Reactors: Past, Present and Future. In: *Journal of Process Control* 14 (2004), Nr. 7, S. 795–805

[Sco04] Scott, S. H.: Optimal Feedback Control and the Neural Basis of Volitional Motor Control. In: *Nature Reviews Neuroscience* 5 (2004), Nr. 7, S. 532–546

[SDM97] Schultz, W.; Dayan, P.; Montague, R.: A Neural Substrate of Prediction and Reward. In: *Science* 275 (1997), Nr. 5306, 1593–1599

[SHM+16] Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; Dieleman, S.; Grewe, D.; Nham, J.; Kalchbrenner, N.; Sutskever, I.; Lillicrap, T.; Leach, M.; Kavukcuoglu, K.; Graepel, T.; Hassabis, D.: Mastering the Game of Go with Deep Neural Networks and Tree Search. In: *Nature* 529-503 (2016), Nr. 7587, 484–503

[SLH+14] Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M.: Deterministic Policy Gradient Algorithms. In: *Proceedings of the 31st International Conference on Machine Learning*, 2014

[SLW17] Song, R.; Lewis, F. L.; Wei, Q.: Off-Policy Integral Reinforcement Learning Method to Solve Nonlinear Continuous-Time Multiplayer Nonzero-Sum Games. In: *IEEE Transactions on Neural Networks and Learning Systems* 28 (2017), Nr. 3, S. 704–713

[SMC+11] Stockar, S.; Marano, V.; Canova, M.; Rizzoni, G.; Guzzella, L.: Energy-Optimal Control of Plug-in Hybrid Electric Vehicles for Real-World Driving Cycles. In: *IEEE Transactions on Vehicular Technology* 60 (2011), Nr. 7, S. 2949–2962

[Spi68] Spiegel, M. R.: *Mathematical Handbook of Formulas and Tables.* New York, St. Louis: McGraw-Hill, 1968

[SSS+17] Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; Chen, Y.; Lillicrap, T.; Hui, F.; Sifre, L.; van den Driessche, G.; Graepel, T.; Hassabis, D.: Mastering the Game of Go Without Human Knowledge. In: *Nature* 550 (2017), Nr. 7676, 354–371

[SSW18] Shi, W.; Song, S.; Wu, C.: High-Level Tracking of Autonomous Underwater Vehicles Based on Pseudo Averaged Q-Learning. In: *IEEE International Conference on Systems, Man, and Cybernetics*, 2018, S. 4138–4143

[Sut88] Sutton, R. S.: Learning to Predict by the Methods of Temporal Differences. In: *Machine Learning* 3 (1988), Nr. 1, S. 9–44

[SWL19] Song, R.; Wei, Q.; Li, Q. F.: *Studies in Systems, Decision and Control.* Bd. 166: *Adaptive Dynamic Programming: Single and Multiple Controllers.* New York, Berlin: Springer, 2019

[Tao03] Tao, G.: *Adaptive Control Design and Analysis.* Hoboken: John Wiley and Sons Inc., 2003

[TBBH10] Teleke, S.; Baran, M. E.; Bhattacharya, S.; Huang, A. Q.: Optimal Control of Battery Energy Storage for Wind Farm Dispatching. In: *IEEE Transactions on Energy Conversion* 25 (2010), Nr. 3, S. 787–794

[TCTH19] Tang, D.; Chen, L.; Tian, Z. F.; Hu, E.: Modified Value-Function-Approximation for Synchronous Policy Iteration with Single-Critic Configuration for Nonlinear Optimal Control. In: *International Journal of Control* 94 (2019), Nr. 5, S. 1–13

[Tho11] Thorndike, E. L.: *Animal Intelligence: Experimental Studies.* New York: The Macmillan Company, 1911

[Tod04] Todorov, E.: Optimality Principles in Sensorimotor Control. In: *Nature Neuroscience* 7 (2004), Nr. 9, S. 907–915

[Vam15] Vamvoudakis, K. G.: Non-Zero Sum Nash Q-Learning for Unknown Deterministic Continuous-Time Linear Systems. In: *Automatica* 61 (2015), S. 274–281

[Vam16] Vamvoudakis, K. G.: Optimal Trajectory Output Tracking Control with a Q-Learning Algorithm. In: *American Control Conference (ACC)*, 2016, S. 5752–5757

[Vam17] Vamvoudakis, K. G.: Q-Learning for Continuous-Time Linear Systems: A Model-Free Infinite Horizon Optimal Control Approach. In: *Systems & Control Letters* 100 (2017), S. 14–20

[van97] van Nieuwstadt, M. J.: *Trajectory Generation for Nonlinear Control Systems.* Pasadena, California Institute of Technology, Dissertation, 1997

[van12] van Hasselt, H.: Reinforcement Learning in Continuous State and Action Spaces. In: Wiering, M. (Hrsg.); van Otterlo, M. (Hrsg.): *Reinforcement Learning: State-of-the-Art.* Berlin, Heidelberg: Springer, 2012, S. 207–251

[VL09] Vrabie, D.; Lewis, F.: Neural Network Approach to Continuous-Time Direct Adaptive Optimal Control for Partially Unknown Nonlinear Systems. In: *Neural Networks* 22 (2009), Nr. 3, S. 237–246

[VL10] Vamvoudakis, K. G.; Lewis, F. L.: Online Actor-Critic Algorithm to Solve the Continuous-Time Infinite Horizon Optimal Control Problem. In: *Automatica* 46 (2010), Nr. 5, S. 878–888

[VL11] Vamvoudakis, K. G.; Lewis, F. L.: Multi-Player Non-Zero-Sum Games: Online Adaptive Learning Solution of Coupled Hamilton-Jacobi Equations. In: *Automatica* 47 (2011), Nr. 8, S. 1556–1569

[VLV13] Vrabie, D. L.; Lewis, F. L.; Vamvoudakis, K. G.: *IET Control Engineering.* Bd. 81: *Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles.* London: Institution of Electrical Engineers, 2013

[VMH16] Vamvoudakis, K. G.; Miranda, M. F.; Hespanha, J. P.: Asymptotically Stable Adaptive-Optimal Control Algorithm With Saturating Actuators and Relaxed Persistence of Excitation. In: *IEEE Transactions on Neural Networks and Learning Systems* 27 (2016), Nr. 11, S. 2386–2398

[VMKL17] Vamvoudakis, K. G.; Modares, H.; Kiumarsi, B.; Lewis, F. L.: Game Theory-Based Control System Algorithms with Real-Time Reinforcement Learning: How to Solve Multiplayer Games Online. In: *IEEE Control Systems Magazine* 37 (2017), Nr. 1, S. 33–52

[VPAKL09] Vrabie, D.; Pastravanu, O.; Abu-Khalaf, M.; Lewis, F. L.: Adaptive Optimal Control for Continuous-Time Linear Systems Based on Policy Iteration. In: *Automatica* 45 (2009), Nr. 2, S. 477–484

[VVL09b] Vrabie, D.; Vamvoudakis, K.; Lewis, F.: Adaptive Optimal Controllers Based on Generalized Policy Iteration in a Continuous-Time Framework. In: *17th Mediterranean Conference on Control and Automation*, 2009, S. 1402–1409

[Wan20] Wang, D.: Robust Policy Learning Control of Nonlinear Plants With Case Studies for a Power System Application. In: *IEEE Transactions on Industrial Informatics* 16 (2020), Nr. 3, S. 1733–1741

[Wat13] Watson, J. B.: Psychology as the Behaviorist Views It. In: *Psychological Review* 20 (1913), Nr. 2, S. 158–177

[Wat89] Watkins, C. J. C. H.: *Learning from Delayed Rewards*, King's College, Cambridge, Dissertation, 1989

[WD92] Watkins, C. J.; Dayan, P.: Q-Learning. In: *Machine Learning* 8 (1992), S. 279–292

[Wei85] Weierstrass, K.: Über die analytische Darstellbarkeit sogenannter willkürlicher Functionen einer reellen Veränderlichen. In: *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin.* 1885, S. 633–639

[Wer77] Werbos, P. J.: Advanced Forecasting Methods for Global Crisis Warning and Models of Intelligence. In: *General Systems* Bd. XXII, 1977, S. 25–38

[Wer92] Werbos, P. J.: Approximate Dynamic Programming for Real-Time Control and Neural Modeling. In: *Handbook of Intelligent Control* (1992), S. 493–526

[Wer99] Werbos, P. J.: Stable Adaptive Control Using New Critic Designs. In: *9th Workshop on Virtual Intelligence/Dynamic Neural Networks.* 1999

[Wer13] Werbos, P. J.: Reinforcement Learning and Approximate Dynamic Programming (RLADP) – Foundations, Common Misconceptions and Challenges Ahead. In: Lewis, F. L. (Hrsg.); Liu, D. (Hrsg.): *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control.* Piscataway, Hoboken: John Wiley and Sons Inc., 2013, S. 3–30

[WHL17] Wang, D.; He, H.; Liu, D.: Adaptive Critic Nonlinear Robust Control: A Survey. In: *IEEE Transactions on Cybernetics* 47 (2017), Nr. 10, S. 3429–3451

[WHQ20] Wang, D.; Ha, M.; Qiao, J.: Self-Learning Optimal Regulation for Discrete-Time Nonlinear Systems Under Event-Driven Formulation. In: *IEEE Transactions on Automatic Control* 65 (2020), Nr. 3, S. 1272–1279

[WL12] Wu, H.-N.; Luo, B.: Neural Network Based Online Simultaneous Policy Update Algorithm for Solving the HJI Equation in Nonlinear $H\infty$ Control. In: *IEEE Transactions on Neural Networks and Learning Systems* 23 (2012), Nr. 12, S. 1884–1895

[WLL14] Wang, D.; Liu, D.; Li, H.: Policy Iteration Algorithm for Online Design of Robust Control for a Class of Continuous-Time Nonlinear Systems. In: *IEEE Transactions on Automation Science and Engineering* 11 (2014), Nr. 2, S. 627–632

[WLLS17] Wei, Q.; Liu, D.; Lin, Q.; Song, R.: Discrete-Time Optimal Control via Local Policy Iteration Adaptive Dynamic Programming. In: *IEEE Transactions on Cybernetics* 47 (2017), Nr. 10, S. 3367–3379

[WLMZ18] Wang, D.; Liu, D.; Mu, C.; Zhang, Y.: Neural Network Learning and Robust Stabilization of Nonlinear Systems With Dynamic Uncertainties. In: *IEEE Transactions on Neural Networks and Learning Systems* 29 (2018), Nr. 4, S. 1342–1351

[WLZZ16] Wang, D.; Liu, D.; Zhang, Q.; Zhao, D.: Data-Based Adaptive Critic Designs for Nonlinear Robust Optimal Control with Uncertain Dynamics. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 46 (2016), Nr. 11, S. 1544–1555

[WSH+16] Wang, Z.; Schaul, T.; Hessel, M.; van Hasselt, H.; Lanctot, M.; de Freitas, N.: Dueling Network Architectures for Deep Reinforcement Learning. In: *Proceedings of the 33rd International Conference on Machine Learning* Bd. 48, 2016, 1995–2003

[WXL+14] Wang, J.; Xu, X.; Liu, D.; Sun, Z.; Chen, Q.: Self-Learning Cruise Control Using Kernel-Based Least Squares Policy Iteration. In: *IEEE Transactions on Control Systems Technology* 22 (2014), Nr. 3, S. 1078–1087

[WY18] Wang, J.-S.; Yang, G.-H.: Output-Feedback Control of Unknown Linear Discrete-Time Systems With Stochastic Measurement and Process Noise via Approximate Dynamic Programming. In: *IEEE Transactions on Cybernetics* 48 (2018), Nr. 7, S. 1977–1988

[WZL09] Wang, F.-Y.; Zhang, H.; Liu, D.: Adaptive Dynamic Programming: An Introduction. In: *IEEE Computational Intelligence Magazine* 4 (2009), Nr. 2, S. 39–47

[WZL16] Wang, T.; Zhang, H.; Luo, Y.: Infinite-Time Stochastic Linear Quadratic Optimal Control for Unknown Discrete-Time Systems Using Adaptive Dynamic Programming Approach. In: *Neurocomputing* 171 (2016), S. 379–386

[WZLD15] Wang, B.; Zhao, D.; Li, C.; Dai, Y.: Design and Implementation of an Adaptive Cruise Control System Based on Supervised Actor-Critic Learning. In: *5th International Conference on Information Science and Technology*, 2015, S. 243–248

[XZLJ16] Xiao, G.; Zhang, H.; Luo, Y.; Jiang, H.: Data-Driven Optimal Tracking Control for a Class of Affine Non-Linear Continuous-Time Systems with Completely Unknown Dynamics. In: *IET Control Theory & Applications* 10 (2016), Nr. 6, S. 700–710

[YDZY20] Yang, Y.; Dong, J.; Zhang, S.; Yin, Y.: Data-Driven Nonzero-Sum Game for Discrete-Time Systems Using Off-Policy Reinforcement Learning. In: *IEEE Access* 4 (2020), Nr. 8, S. 14074–14088

[YHL13] Yang, X.; Huang, Y.; Liu, D.: Neural-Network-Based Online Optimal Control for Uncertain Non-Linear Continuous-Time Systems with Control Constraints. In: *IET Control Theory & Applications* 7 (2013), Nr. 17, S. 2037–2047

[YLLL16] Yang, X.; Liu, D.; Luo, B.; Li, C.: Data-Based Robust Adaptive Control for a Class of Unknown Nonlinear Constrained-Input Systems via Integral Reinforcement Learning. In: *Information Sciences* 369 (2016), S. 731–747

[YSH+17] Yu, R.; Shi, Z.; Huang, C.; Li, T.; Ma, Q.: Deep Reinforcement Learning Based Optimal Trajectory Tracking Control of Autonomous Underwater Vehicle. In: *36th Chinese Control Conference*, 2017, S. 4958–4965

[YWM+19] Yang, Y.; Wang, L.; Modares, H.; Ding, D.; Yin, Y.; Wunsch, D.: Data-Driven Integral Reinforcement Learning for Continuous-Time Non-Zero-Sum Games. In: *IEEE Access* 7 (2019), S. 82901–82912

[ZCL13] Zhang, H.; Cui, L.; Luo, Y.: Near-Optimal Control for Nonzero-Sum Differential Games of Continuous-Time Nonlinear Systems Using Single-Network ADP. In: *IEEE Transactions on Cybernetics* 43 (2013), Nr. 1, S. 206–216

[ZCZL11] Zhang, H.; Cui, L.; Zhang, X.; Luo, Y.: Data-Driven Robust Approximate Optimal Tracking Control for Unknown General Nonlinear Systems Using Adaptive Dynamic Programming Method. In: *IEEE Transactions on Neural Networks* 22 (2011), Nr. 12, S. 2226–2236

[ZD98] Zhou, K.; Doyle, J. C.: *Essentials of Robust Control.* Upper Saddle River: Prentice Hall, 1998

[ZDJ14] Zargarzadeh, H.; Dierks, T.; Jagannathan, S.: Adaptive Neural Network-Based Optimal Control of Nonlinear Continuous-Time Systems in Strict-Feedback Form. In: *International Journal of Adaptive Control and Signal Processing* 28 (2014), S. 305–324

[ZDJ15] Zargarzadeh, H.; Dierks, T.; Jagannathan, S.: Optimal Control of Nonlinear Continuous-Time Systems in Strict-Feedback Form. In: *IEEE Transactions on Neural Networks and Learning Systems* 26 (2015), Nr. 10, S. 2535–2549

[Zim84] Zimmer, K.: Determination of Proper Output Variables for the Decoupling of Nonlinear Systems. In: *Electronics Letters* 20 (1984), Nr. 25–26, S. 1052–1053

[ZQJL14] Zhang, H.; Qin, C.; Jiang, B.; Luo, Y.: Online Adaptive Policy Learning Algorithm for H-infinity State Feedback Control of Unknown Affine Nonlinear Discrete-Time Systems. In: *IEEE Transactions on Cybernetics* 44 (2014), Nr. 12, S. 2706–2718

[ZWL08] Zhang, H.; Wei, Q.; Luo, Y.: A Novel Infinite-Time Optimal Tracking Control Scheme for a Class of Discrete-Time Nonlinear Systems via the Greedy HDP Iteration Algorithm. In: *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics* 38 (2008), Nr. 4, S. 937–942

[ZZWZ16] Zhao, D.; Zhang, Q.; Wang, D.; Zhu, Y.: Experience Replay for Optimal Control of Nonzero-Sum Game Systems with Unknown Dynamics. In: *IEEE Transactions on Cybernetics* 46 (2016), Nr. 3, S. 854–865

[ZZXS17] Zhang, K.; Zhang, H.; Xiao, G.; Su, H.: Tracking Control Optimization Scheme of Continuous-Time Nonlinear System via Online Single Network Adaptive Critic Design Method. In: *Neurocomputing* 251 (2017), S. 127–135

Eigene Veröffentlichungen und Tagungsbeiträge

[BKIH21] Bührle, E.; Köpf, F.; Inga, J.; Hohmann, S.: Adaptive Optimal Trajectory Tracking Control of Continuous-Time Systems. In: *European Control Conference (ECC)*, 2021, S. 1987–1994

[IBKH20] Inga, J.; Bischoff, E.; Köpf, F.; Hohmann, S.: Inverse Dynamic Games Based on Maximum Entropy Inverse Reinforcement Learning. In: *arXiv preprint 1911.07503v2* (2020), S. 1–8

[IKFH17] Inga, J.; Köpf, F.; Flad, M.; Hohmann, S.: Individual Human Behavior Identification Using an Inverse Reinforcement Learning Method. In: *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2017, S. 99–104

[IRKH17] Inga, J.; Rothfuss, S.; Köpf, F.; Hohmann, S.: *Inverse Optimierung in linear-quadratischen dynamischen Spielen.* Günzburg, 2017 (GMA-Fachausschuss 1.50 Workshop „Grundlagen vernetzter Systeme“)

[KEFH18] Köpf, F.; Ebbert, S.; Flad, M.; Hohmann, S.: Adaptive Dynamic Programming for Cooperative Control with Incomplete Information. In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2018, S. 2632–2638

[KFH19] Köpf, F.; Flad, M.; Hohmann, S.: *Kooperative Optimalregelung unter eingeschränkter Kommunikation und Information.* Günzburg, 2019 (GMA-Fachausschuss 1.50 Workshop „Grundlagen vernetzter Systeme“)

[KIB+19] Kastner, A.; Inga, J.; Blauth, T.; Köpf, F.; Flad, M.; Hohmann, S.: Model-Based Control of a Large-Scale Ball-on-Plate System With Experimental Validation. In: *IEEE International Conference on Mechatronics (ICM)*, 2019, S. 257–262

[KIR+17] Köpf, F.; Inga, J.; Rothfuss, S.; Flad, M.; Hohmann, S.: Inverse Reinforcement Learning for Identification in Linear-Quadratic Dynamic Games. In: *IFAC-PapersOnLine* 50 (2017), Nr. 1, S. 14902–14908

[KKBH23] Karg, P.; Köpf, F.; Braun, C. A.; Hohmann, S.: Excitation for Adaptive Optimal Control of Nonlinear Systems in Differential Games. In: *IEEE Transactions on Automatic Control* (2023)

[KKF+20] Kuhn, E.; Kusmartsev, F.; Flad, M.; Hohmann, S.; Köpf, F.: *Bewegungsabhängiges Stabilisierungsunterstützungssystem: Patent DE 10 2019 210 232.* 2020

[KKIH21] Köpf, F.; Kille, S.; Inga, J.; Hohmann, S.: Adaptive Optimal Trajectory Tracking Control Applied to a Large-Scale Ball-on-Plate System. In: *IEEE American Control Conference (ACC)*, 2021, S. 1352–1357

[KNFH20] Köpf, F.; Nitsch, A.; Flad, M.; Hohmann, S.: Partner Approximating Learners (PAL): Simulation-Accelerated Learning with Explicit Partner Modeling in Multi-Agent Domains. In: *IEEE International Conference on Control, Automation and Robotics (ICCAR)*, 2020, S. 746–752

[KPRH20] Köpf, F.; Puccetti, L.; Rathgeber, C.; Hohmann, S.: Reinforcement Learning for Speed Control with Feedforward to Track Velocity Profiles in a Real Vehicle. In: *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2020, S. 1–8

[KRP+20] Köpf, F.; Ramsteiner, S.; Puccetti, L.; Flad, M.; Hohmann, S.: Adaptive Dynamic Programming for Model–Free Tracking of Trajectories with Time–Varying Parameters. In: *International Journal of Adaptive Control and Signal Processing* 34 (2020), Nr. 7, S. 839–856

[KTFH20] Köpf, F.; Tesfazgi, S.; Flad, M.; Hohmann, S.: Deep Decentralized Reinforcement Learning for Cooperative Control. In: *IFAC-PapersOnLine* 53 (2020), Nr. 2, S. 1555–1562

[KWFH20] Köpf, F.; Westermann, J.; Flad, M.; Hohmann, S.: Adaptive Optimal Control for Reference Tracking Independent of Exo-System Dynamics. In: *Neurocomputing* 405 (2020), S. 173–185

[LKS+18] Lemmer, M.; Köpf, F.; Schwab, S.; Flad, M.; Hohmann, S.: Modeling of Human-Centered Cooperative Control by Means of Tracking in Discrete-Time Linear-Quadratic Differential Games. In: *IEEE International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, 2018, S. 156–161

[PKRH20] Puccetti, L.; Köpf, F.; Rathgeber, C.; Hohmann, S.: Speed Tracking Control Using Online Reinforcement Learning in a Real Car. In: *IEEE International Conference on Control, Automation and Robotics (ICCAR)*, 2020, S. 392–399

[RIK+17] Rothfuss, S.; Inga, J.; Köpf, F.; Flad, M.; Hohmann, S.: Inverse Optimal Control for Identification in Non-Cooperative Differential Games. In: *IFAC-PapersOnLine* 50 (2017), Nr. 1, S. 14909–14915

Betreute studentische Arbeiten

[Bla18] Blauth, T.: *Kamerabasierte Messung der Ballposition eines Ball-auf-Platte-Systems*, Fakultät für Elektrotechnik und Informationstechnik, Karlsruher Institut für Technologie, Bachelorarbeit, 2018

[Bra18] Braun, C. A.: *Entwicklung und Analyse Reinforcement-Learning-basierter Regler für Differentialspiele mit unbekanntem Kooperationspartner*, Fakultät für Elektrotechnik und Informationstechnik, Karlsruher Institut für Technologie, Masterarbeit, 2018

[Büh20] Bührle, E.: *Reinforcement Learning zur adaptiven Zustandsfolgeregelung zeitkontinuierlicher Systeme*, Fakultät für Elektrotechnik und Informationstechnik, Karlsruher Institut für Technologie, Masterarbeit, 2020

[Den17] Dengler, K.: *Implementierung und Untersuchung eines Expectation-Maximization-Algorithmus zur Trajektorienplanung in endlichen Zustandsräumen*, Fakultät für Elektrotechnik und Informationstechnik, Karlsruher Institut für Technologie, Bachelorarbeit, 2017

[Ebb17] Ebbert, S.: *Bewertung von Methoden zur Regelung kooperativer Systeme mit unbekanntem Handlungspartner*, Fakultät für Elektrotechnik und Informationstechnik, Karlsruher Institut für Technologie, Masterarbeit, 2017

[Gaj18] Gajek, S.: *Development of a Solution Concept for Nonlinear Cooperative Control Problems by Means of Dynamic Game Theory and Methods of Reinforcement Learning*, Fakultät für Elektrotechnik und Informationstechnik, Karlsruher Institut für Technologie, Masterarbeit, 2018

[Kar19] Karg, P.: *Strukturierte Anregung Reinforcement-Learning-basierter kooperativer Regler*, Fakultät für Elektrotechnik und Informationstechnik, Karlsruher Institut für Technologie, Masterarbeit, 2019

[Kil20] Kille, S.: *Anwendung Reinforcement-Learning-basierter Trajektorienfolgeregler auf ein Ball-auf-Platte-System*, Fakultät für Elektrotechnik und Informationstechnik, Karlsruher Institut für Technologie, Masterarbeit, 2020

[Lem18] Lemmer, M.: *Entwurf und Analyse zeitdiskreter Linear-Quadratischer Zwei-Spieler-Folgeregelungen*, Fakultät für Elektrotechnik und Informationstechnik, Karlsruher Institut für Technologie, Masterarbeit, 2018

[Nit18] Nitsch, A.: *Entwicklung kooperativer Regler basierend auf Identifikationsmethoden und Deep Reinforcement Learning*, Fakultät für Elektrotechnik und Informationstechnik, Karlsruher Institut für Technologie, Masterarbeit, 2018

[Ram19] Ramsteiner, S.: *Parameterbasierte Folgeregelung unter Verwendung von Reinforcement-Learning-Methoden*, Fakultät für Elektrotechnik und Informationstechnik, Karlsruher Institut für Technologie, Masterarbeit, 2019

[Tes19] TESFAZGI, S.: *Deep Decentralised Reinforcement Learning for Cooperative Control*, Fakultät für Elektrotechnik und Informationstechnik, Karlsruher Institut für Technologie, Masterarbeit, 2019

[Wes18] WESTERMANN, J.: *Optimale Folgeregelung für unbekannte lineare Systeme basierend auf Q-Learning mit gleitendem Horizont*, Fakultät für Elektrotechnik und Informationstechnik, Karlsruher Institut für Technologie, Masterarbeit, 2018

[Wie19] WIECZOREK, D. H.: *Untersuchung und Modellierung menschlicher Lernprozesse in haptisch gekoppelten kooperativen Szenarien*, Fakultät für Elektrotechnik und Informationstechnik, Karlsruher Institut für Technologie, Masterarbeit, 2019

Karlsruher Beiträge zur Regelungs- und Steuerungstechnik (ISSN 2511-6312)
Institut für Regelungs- und Steuerungssysteme

Band 01 Diehm, Gunter
Identifikation des menschlichen Bewegungsverhaltens
auf der Basis von Primitiven.
ISBN 978-3-7315-0608-9

Band 02 Flad, Michael
Kooperative Regelungskonzepte auf Basis der Spieltheorie
und deren Anwendung auf Fahrerassistenzsysteme.
ISBN 978-3-7315-0610-2

Band 03 Eckert, Marius
Modellbasierte Identifikation fraktionaler Systeme
und ihre Anwendung auf die Lithium-Ionen-Zelle.
ISBN 978-3-7315-0690-4

Band 04 Krebs, Stefan
Intervallbeobachter für lineare parametervariante Systeme
und deren Anwendung auf die Asynchronmaschine.
ISBN 978-3-7315-0857-1

Band 05 Kaspar, Stephan
Fahrdynamikuntersuchungen eines Elektrofahrzeugs
mit Einzelrad-Hinterradantrieb.
ISBN 978-3-7315-0916-5

Band 06 Sauter, Patrick S.
Modellierung und zentrale prädiktive Regelung
von multimodalen Energieverteilnetzen.
ISBN 978-3-7315-0963-9

Band 07 Kupper, Martin
Verteilte Zustandsschätzung fraktionaler Systeme und
ihre Anwendung auf Lithium-Ionen-Batteriesysteme.
ISBN 978-3-7315-0971-4

Band 08 Merkert, Lennart
Optimal Scheduling of Combined Heat and Power Generation
Considering Heating Grid Dynamics.
ISBN 978-3-7315-1056-7

Die Bände sind unter www.ksp.kit.edu als PDF frei verfügbar oder als Druckausgabe bestellbar.

Band 09 Ludwig, Julian
Automatisierte kooperative Transition einer Regelungsaufgabe zwischen Mensch und Maschine am Beispiel des hochautomatisierten Fahrens.
ISBN 978-3-7315-1069-7

Band 10 Inga Charaja, Juan Jairo
Inverse Dynamic Game Methods for Identification of Cooperative System Behavior.
ISBN 978-3-7315-1080-2

Band 11 Schnurr, Christoph Xaver
Ein Verfahren zur lexikographischen modellprädiktiven Regelung mit der Anwendung auf eine permanenterregte Synchronmaschine.
ISBN 978-3-7315-1095-6

Band 12 Schwab, Stefan
Guaranteed Verification of Dynamic Systems.
ISBN 978-3-7315-0965-3

Band 13 Pfeifer, Martin
Automated Model Generation and Observer Design for Interconnected Systems: A Port-Hamiltonian Approach.
ISBN 978-3-7315-1135-9

Band 14 König, Alexander
Absicherung hochautomatisierten Fahrens durch passiven virtuellen Dauerlauftest.
ISBN 978-3-7315-1141-0

Band 15 Stark, Oliver
Parameter- und Ordnungsidentifikation von fraktionalen Systemen mit einer Anwendung auf eine Lithium-Ionen-Batteriezelle.
ISBN 978-3-7315-1187-8

Band 16 Köpf, Florian
Adaptive Dynamic Programming: Solltrajektorienfolgeregelung und Konvergenzbedingungen.
ISBN 978-3-7315-1193-9



16

Adaptive Dynamic Programming (ADP) steht als vielversprechendes und zukunftsorientiertes regelungstechnisches Werkzeug im Fokus der aktuellen Forschung. Allerdings existieren hierfür bislang weder flexibel einsetzbare, mit dem ADP-Mechanismus kompatible Solltrajektoriendarstellungen noch theoretische Untersuchungen hinsichtlich einer geeigneten Systemanregung zur Sicherstellung der Konvergenz. Die vorliegende Arbeit schließt diese Lücken: Zum einen werden erstmals zeitdiskrete und zeitkontinuierliche Methoden präsentiert und analysiert, die flexible Solltrajektoriendarstellungen in ADP-Ansätze integrieren. Die explizite Abhängigkeit der vorgestellten, neuartigen Value- bzw. Q-Function und des darauf basierenden gelernten Regelgesetzes von Trajektorienparametern, die den aktuellen Sollverlauf repräsentieren, ermöglicht eine variable Vorgabe der Solltrajektorie zur Laufzeit. Zum anderen werden erstmalig theoretische Bedingungen an den Systemzustand hergeleitet, die sicherstellen, dass eine für die Konvergenz der Adaption zentrale Anregungseigenschaft erfüllt ist. Verbleibende Freiheitsgrade erlauben zudem die Berücksichtigung anwendungsspezifischer Anforderungen bei der Systemanregung. Die theoretischen Aussagen werden in Simulationen bestätigt. Erste reale Anwendungen der vorgestellten adaptiven optimalen Trajektorienfolgeregelungsmethoden offenbaren schließlich das Potenzial dieser Ansätze. Flexible und effiziente Regler, die aufgrund der Berücksichtigung des Solltrajektorienverlaufs vorausschauend agieren, können ohne aufwendige Modellbildung aus realen Messdaten erlernt werden und sind zudem bisherigen Ansätzen bezüglich ihrer Performanz überlegen.

ISBN 978-3-7315-1193-9
9 783731 511939 >

ISSN 2511-6312
ISBN 978-3-7315-1193-9

Gedruckt auf FSC-zertifiziertem Papier